

THESES

présentées

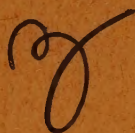
à la Faculté des Sciences de l'Université de Nantes

pour obtenir

le GRADE de DOCTEUR - INGENIEUR

par

Gérard VEILHAN



1^{ère} THESE : Approximations des fonctions usuelles en vue du calcul sur ordinateur.
Application à l'I.B.M. 650 Standard.

2^{ème} THESE : *Propositions données par la Faculté.*

Soutenues le 29 Juin 1965 devant la Commission d'Examen

M. BRILLOUET Président

M. PAUC
M. RAVILLY } Examinateurs

THESES

présentées

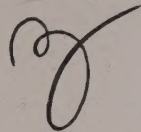
à la Faculté des Sciences de l'Université de Nantes

pour obtenir

le GRADE de DOCTEUR - INGENIEUR

par

Gérard VEILHAN



1^{ère} THESE : Approximations des fonctions usuelles en vue du calcul sur ordinateur.
Application à l'I.B.M. 650 Standard.

2^{ème} THESE : *Propositions données par la Faculté.*

Soutenues le 29 Juin 1965 devant la Commission d'Examen

M. BRILLOUET Président

M. PAUC

M. RAVILLY

} Examinateurs

 FACULTE DES SCIENCES

Doyen M. BRILLOUET
Assesseur M. GARRIGUES

Professeurs Titulaires

MM. BRILLOUET	<i>Mathématiques</i>
PAUC	«
PETRESCU	«
RAFFIN	«
PIRONNEAU	<i>Mécanique</i>
RAVILLY	«
BOULLOUD	<i>Physique</i>
COLLET	«
DUNOYER	<i>Chimie</i>
GRAFF	«
M ^{lle} DEBRAUX	<i>Biologie</i>
MM. GARRIGUES	«
GOURINARD	<i>Géologie</i>

Professeurs et Maîtres de Conférences

MM. MOLINARO	<i>Mathématiques</i>
VIVIER	«
VAILLANT	«
BALQUET	<i>Mécanique</i>
AVIGNON	<i>Physique</i>
LERMUSIAUX	«
MEZENCEV	«
GOUREAUX	«
LASKAR	«
MINN	«
MOUTON	«
M ^{lle} OFFRET	«
MM. GOBIN	<i>Chimie</i>
JOLIVET	«
QUINIOU	«
ROUXEL	«
WOJTKOWIAK	«
MARTIN	«
PIERI	<i>Biologie</i>
SELLIER	«
DUPONT	«
OTTMANN	<i>Géologie</i>
FORESTIER	«

Secrétaire

M. PAILLET

Que Monsieur le Professeur BRILLOUET, Doyen de la Faculté des Sciences de Nantes, me permette de lui exprimer ici mes sincères remerciements pour les précieux conseils qu'il a bien voulu me prodiguer au cours de ce travail .

Je remercie vivement Messieurs les Professeurs qui m'ont fait l'honneur de composer le Jury .

Je remercie aussi toutes les personnes qui se sont chargées de la bonne présentation et de l'impression de ce fascicule .

A ma femme .

A la mémoire de notre fils Cyril .

A mes parents et beaux - parents .

TABLE DES MATIERES

PAGES

INTRODUCTION

PREMIERE PARTIE : THEORIE DES DIVERSES APPROXIMATIONS

11 - <u>RACINE CARREE</u>	1
111 - Principe	1
112 - Comparaison suivant le choix de la première approximation	6
12 - <u>FONCTIONS TRANSCENDANTES</u>	7
121 - Etude théorique des différentes approximations	7
121.1 - Approximations polynomiales	7
121.11 - Développement en polynômes de Tchebycheff	7
121.12 - Développement en série de puissance et relaxation	12
121.13 - Développement d'Hastings	13
121.2 - Approximation rationnelle	13
122 - Application aux fonctions transcendantes	13
122.1 - <u>Logarithme Népérien</u>	14
122.11 - Calcul numérique des approximations	14
122.111 - Approximations polynomiales	14
122.111.1 - Développement de Tchebycheff	14
122.111.2 - Développement en Série et Relaxation	20
122.111.3 - Développement d'Hastings	22
122.112 - Approximation rationnelle	23
122.12 - Comparaison des approximations	26
122.2 - <u>Exponentielle</u>	27
122.21 - Calcul numérique des approximations	27
122.211 - Approximations polynomiales	28
122.211.1 - Développement de Tchebycheff	28
122.211.2 - Développement en Série et Relaxation	30
122.211.3 - Développement d'Hastings	33
122.212 - Approximation rationnelle	34
122.22 - Comparaison des approximations	37
122.3 - <u>Sinus - Cosinus</u>	39
122.31 - Calcul numérique des approximations	39
122.311 - Approximations polynomiales	39
122.311.1 - Développement de Tchebycheff	39
122.311.2 - Développement en Série et Relaxation	45
122.311.3 - Développement d'Hastings	51

122.312 - Approximation rationnelle	51
122.32 - Comparaison des approximations	54
122.4 - Arc Tangente	56
122.41 - Calcul numérique des approximations	56
122.411 - Approximations polynomiales	56
122.411.1 - Développement de Tchebycheff	56
122.411.2 - Développement en Série et Relaxation	60
122.411.3 - Développement d'Hastings	63
122.412 - Approximation rationnelle	63
122.42 - Comparaison des approximations	65
122.5 - Arc Sinus	66
122.51 - Calcul numérique des approximations	66
122.511 - Développement de Tchebycheff	66
122.512 - Développement d'Hastings	70
122.52 - Comparaison entre le calcul direct de Arc Sinus et le calcul par Arc Tangente	71
SECONDE PARTIE : PROGRAMMATION SUR I.B.M.650 STANDARD	
21 - GENERALITES	73
22 - COMPARAISON DES DIFFERENTES METHODES PROGRAMMEES (SIMPLE PRECISION et DOUBLE PRECISION)	79
23 - NOTICES D'UTILISATION DES SOUS-PROGRAMMES DES BIBLIOTHEQUES	99
231 - Sous-programmes fonctionnels en Simple Précision	99
231.1 - Sous-programmes « Racine Carrée » (Références SFRCE et SFRCK) Utilisation en virgule fixe et virgule flottante	99
231.2 - Sous-programme « Logarithme Népérien » (Référence SFLE) Utilisation en virgule fixe et virgule flottante	104
231.3 - Sous-programmes « Exponentielle » (Références SFEXE et SFEXK) Utilisation en virgule fixe et virgule flottante	105
231.4 - Sous-programme « Sinus-Cosinus » (Référence SFSC) Utilisation en virgule fixe et virgule flottante	109
231.5 - Sous-programme « Arc Tangente » (Référence SFAT) Utilisation en virgule fixe et virgule flottante	111
231.6 - Sous-programme « Arc Sinus » (Référence SFAS) Utilisation en virgule fixe et virgule flottante	113
232 - Sous-programmes fonctionnels en Double Précision	115
232.1 - Sous-programmes « Racine Carrée » (Références DFRCE et DFRCK) Utilisation en virgule flottante	115
232.2 - Sous-programme « Logarithme Népérien » (Référence DFLE) Utilisation en virgule flottante	117
232.3 - Sous-programme « Exponentielle » (Référence DFEX) Utilisation en virgule flottante	118
232.4 - Sous-programme « Sinus-Cosinus » (Référence DFSC) Utilisation en virgule flottante	119
232.5 - Sous-programme « Arc Tangente » (Référence DFAT) Utilisation en virgule flottante	120
CONCLUSION	
123	123
ANNEXES :	
ANNEXE 1 : Notices explicatives des sous-programmes fonctionnels pour I.B.M.650 Standard	125
ANNEXE 2 : Bibliographie	161

INTRODUCTION

Nous nous proposons, dans cette étude, d'établir deux bibliothèques des fonctions élémentaires : l'une en simple précision (8 ou 10 chiffres significatifs suivant la représentation utilisée : virgule flottante ou virgule fixe), l'autre, en double précision (18 chiffres significatifs : nous utiliserons uniquement la virgule flottante). Ces bibliothèques doivent viser à réunir les trois qualités suivantes :

- un encombrement restreint,
- un temps de calcul réduit,
- une précision aussi grande que possible .

Nous nous limiterons aux principales méthodes données par E.G. KOGBETLIANTZ, c'est-à-dire :

- Approximation obtenue par un développement de TCHEBYCHEFF limité à une somme partielle.
- Approximation obtenue par un développement en série de MAC-LAURIN limité à une somme partielle, avec relaxation des derniers termes .
- Approximation rationnelle obtenue par la série de DARBOUX ou par le développement limité en fraction continue .

Les résultats obtenus par KOGBETLIANTZ concernent essentiellement les machines « purement binaires » . Nos bibliothèques étant destinées au 650, nous avons repris les calculs avec les modifications imposées par une machine décimale.

Dans l'étude théorique des différentes approximations, nous poussons le développement jusqu'à obtenir une erreur de méthode inférieure à une limite donnée ; nous connaissons alors le nombre de termes à calculer et, par suite, le nombre d'opérations ; de ce nombre dépendent naturellement l'encombrement et le temps de calcul ; c'est pourquoi une comparaison rapide nous permet déjà de procéder à une première élimination. Notre choix définitif se basera sur la comparaison des résultats de programmation qui, en tenant compte de la réduction d'intervalle, du changement de variable, de la nature des opérations et de la perte de précision, nous indiquent avec sûreté, l'encombrement , le temps et la précision obtenus. Aussi, avons-nous divisé notre étude en deux parties. Nous intitulerons la première « partie théorique » puisqu'elle reste indépendante du matériel utilisé, pourvu

qu'il s'agisse toutefois d'une machine décimale . Elle consiste en un résumé des méthodes avec leur application aux différentes fonctions, c'est-à-dire, le calcul des coefficients de l'approximation, la détermination de l'erreur de méthode et les formules numériques définitives retenues en vue de la programmation. La seconde sera la « partie pratique » : elle concerne la programmation effectuée sur IBM 650 Standard . Elle comporte, outre quelques généralités explicatives, l'étude comparative de l'encombrement, du temps de calcul et de la précision des sous-programmes fonctionnels, ainsi que le choix des meilleurs ; nous insisterons dans cette partie sur les causes de perte de précision lors de la programmation ; nous y joignons les notices d'utilisation des sous-programmes retenus pour les bibliothèques .

Une partie complémentaire (Annexes) donne, d'une part, le détail des sous-programmes retenus, d'autre part, une bibliographie .

PREMIERE PARTIE



THEORIE DES DIVERSES APPROXIMATIONS

Les fonctions classiques sont de deux sortes : les unes sont calculées par la résolution d'une équation de type $f(x) = 0$; c'est le cas de la « Racine Carrée ». Les autres sont les « fonctions transcendentes » ; leur calcul est effectué à l'aide d'une approximation .

11 - RACINE CARREE

Il s'agit de résoudre l'équation $x^2 - a = 0$. Nous le faisons par la méthode de Newton .

111 - Principe

Nous envisageons un intervalle $[\alpha, \beta]$ tel que $\sqrt{a}$ soit la seule solution de l'équation dans cet intervalle. Nous choisissons une valeur x_0 ($\alpha < x_0 < \beta$) approchée de la racine par excès. La tangente à la courbe $f(x) = x^2 - a$, menée par le point P de coordonnées x_0 et $f(x_0)$ coupe l'axe des x au point d'abscisse x_1 . Comme la courbe $y = f(x)$ tourne sa concavité vers les y positifs, la tangente en P est tout entière au-dessous de la courbe et, par conséquent, donne une valeur x_1 supérieure à la racine. De plus, x_0 ayant été choisi par excès, x_1 est inférieur à x_0 et représente donc une valeur plus approchée de la racine. x_1 est donné par l'équation de la tangente dans laquelle $y = 0$:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

REMARQUE : $\sqrt{a}$ n'étant pas racine double, $f'(\sqrt{a})$ n'est pas nul .

Nous pouvons poursuivre la méthode à partir de x_1 , puis de x_2 ... Nous obtenons une suite de valeurs approchées, toutes supérieures à la racine et s'en rapprochant de plus en plus. Ces valeurs sont définies par la formule de récurrence :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{x_n^2 - a}{2x_n} = \frac{1}{2} \left(x_n + \frac{a}{x_n} \right)$$

Si nous voulons que l'erreur porte sur le p^e chiffre de $\sqrt{a}$, nous arrêtons les itérations dès que x_{n+1} et x_n ont $(p-1)$ chiffres communs.

Connaissant l'écart $(x_0 - \sqrt{a})$, nous pouvons déterminer une limite supérieure de l'écart $(x_1 - \sqrt{a})$. Au voisinage de $\sqrt{a}$, le développement en série de Taylor de la fonction $f(x)$ est :

$$f(x) = f(\sqrt{a}) = 0 = f(x_0) + (\sqrt{a} - x_0) f'(x_0) + \frac{(\sqrt{a} - x_0)^2}{2} f''(\xi)$$

Nous en déduisons après division par $f'(x_0)$ ($\neq 0$)

$$x_1 - \sqrt{a} = \frac{f''(\xi)}{2f'(x_0)} (x_0 - \sqrt{a})^2$$

Si B est une borne supérieure de $|f''(\xi)|$ dans l'intervalle où nous nous plaçons :

$$(x_1 - \sqrt{a}) < \frac{B}{2|f'(x_0)|} (x_0 - \sqrt{a})^2$$

L'erreur relative à x_1 est de l'ordre du carré de l'erreur relative à x_0 . Il en est de même quand nous passons de x_n à x_{n+1} : à chaque itération, le nombre de chiffres exacts double sensiblement. Pour réduire le nombre d'itérations, il suffit de trouver une valeur x_0 suffisamment proche de la racine.

REMARQUE : Nous choisissons un intervalle $[\alpha, \beta]$ positif (séparation de la racine positive) et se présentant sous la forme $[10^{p-1}; 10^p]$ afin de le réduire par un simple décalage de la virgule de l'argument.

$$\text{III. - 1} \quad x_0 = \frac{1}{2} \left(\frac{a}{k} + k \right)$$

La valeur x_0 ainsi choisie est bien supérieure à $\sqrt{a}$ car $\frac{a}{k} + k > 2\sqrt{a}$ peut s'écrire

$$(\sqrt{a} - k)^2 > 0. \text{ La valeur de } k \text{ la plus employée}$$

est l'unité mais nous pouvons réduire l'écart $(x_0 - \sqrt{a})$ en choisissant k suivant la valeur de l'argument. Par exemple, si « a » est réduit dans l'intervalle $[1, 10[$ nous donnons à k la valeur $\sqrt{s}$, s désignant la partie entière de l'argument. La valeur x_0 est alors plus proche de la racine que celle qui était obtenue avec $k=1$. La valeur $\sqrt{s}$ ne nécessite aucun calcul puisqu'elle peut être fournie par une Table.

$$111 - 2 \quad x_0 = A(B+a - \frac{C}{D+a})$$

Cette expression rationnelle, si les coefficients A, B, C, D , sont bien choisis, permet d'obtenir une valeur x_0 beaucoup plus proche de la racine que toutes les valeurs précédentes.

L'argument est réduit dans l'intervalle $[0,1 ; 1[$. Les coefficients A, B, C, D sont à choisir de façon que l'écart $|x_0 - \sqrt{a}|$ soit aussi petit que possible dans tout l'intervalle. Dans sa conférence du 25 septembre 1956, E.G. Kogbetliantz insiste sur le fait que ces coefficients, comme l'ont montré ses expériences, « ne peuvent être uniques sur tout l'intervalle, mais qu'il faut diviser celui-ci en plusieurs sous-intervalles dans lesquels l'intégrale de la courbure de la courbe $y = \sqrt{x}$ a la même valeur. »

Nous divisons l'intervalle $[0,1 ; 1[$ en trois sous-intervalles. Si i et j sont les bornes intermédiaires, elles sont déterminées par les relations :

$$\int_{0,1}^i (1+4a)^{-3/2} da = \int_i^j (1+4a)^{-3/2} da = \int_j^1 (1+4a)^{-3/2} da$$

Soit :

$$\frac{1}{\sqrt{1,4}} - \frac{1}{\sqrt{1+4i}} = \frac{1}{\sqrt{1+4i}} - \frac{1}{\sqrt{1+4j}} = \frac{1}{\sqrt{1+4j}} - \frac{1}{\sqrt{5}}$$

La résolution de ces deux équations donne :

$$i = 0,24244899$$

$$j = 0,49352054$$

Nous modifions légèrement les valeurs de i et j en les choisissant respectivement égales à 0,23 et 0,529; il en résulte la propriété suivante : les sous intervalles I_2 et I_3 se déduisent respectivement de I_1 et I_2 en les multipliant par 2,3 ,

$$\begin{aligned} \text{d'où} : \quad I_1 & (0,1 ; 0,23) \\ I_2 & (0,23 ; 0,529) \\ I_3 & (0,529 ; 1,2167) \end{aligned}$$

Cette propriété permet d'exprimer les coefficients A_2, B_2, C_2, D_2 relatifs à l_2 et A_3, B_3, C_3, D_3 relatifs à l_3 uniquement en fonction de $\mu = \sqrt{2,3}$ et de A_1, B_1, C_1, D_1 relatifs à l_1 .
La connaissance de A_1, B_1, C_1, D_1 entraîne celle de tous les autres coefficients.

En effet, si « a_1 » appartient à l_1 , $\mu^2 a_1$ appartient à l_2 d'après la propriété ci-dessus. Posons $a_2 = \mu^2 a_1$

$$\sqrt{a_2} = \mu \sqrt{a_1} = \mu A_1 \left[B_1 + \frac{a_2}{\mu^2} - \frac{C_1}{D_1 + \frac{a_2}{\mu^2}} \right]$$

Par identification, nous en déduisons :

$$\begin{aligned} A_2 &= \frac{A_1}{\mu} & A_3 &= \frac{A_2}{\mu} = \frac{A_1}{\mu^2} \\ B_2 &= \mu^2 B_1 & B_3 &= \mu^2 B_2 = \mu^4 B_1 \\ C_2 &= \mu^4 C_1 & C_3 &= \mu^4 C_2 = \mu^8 C_1 \\ D_2 &= \mu^2 D_1 & D_3 &= \mu^2 D_2 = \mu^4 D_1 \end{aligned}$$

Désormais, nous nous plaçons dans l'intervalle l_1 .

L'écart $(x_0 - \sqrt{a_1}) = A_1(B_1 + a_1 - \frac{C_1}{D_1 + a_1}) - \sqrt{a_1}$ peut se mettre sous la forme suivante en introduisant la nouvelle variable $z = \sqrt{a_1}$:

$$\frac{\mathcal{X}(z)}{\mathcal{D}(z)} = \frac{A_1 z^4 - z^3 + A_1(B_1 + D_1)z^2 - D_1 z + A_1(B_1 D_1 - C_1)}{D_1 + z^2}$$

La résolution de l'équation du quatrième degré $\mathcal{X}(z) = 0$ donne quatre solutions. Il y a donc dans l'intervalle l_1 quatre points pour lesquels l'écart $(x_0 - \sqrt{a_1})$ est nul.

Posons : $z_1 = \sqrt{\varphi_1}$, $z_2 = \sqrt{\varphi_2}$, $z_3 = \sqrt{\varphi_3}$, $z_4 = \sqrt{\varphi_4}$ les racines de $\mathcal{X}(z)$, avec $0,1 < \varphi_1 < \varphi_2 < \varphi_3 < \varphi_4 < 0,23$

$$\text{et} \quad S_1 = \sum_1^4 z_i \quad S_2 = \sum_1^4 z_i^2 \quad S_{-1} = \sum_1^4 z_i^{-1} \quad P = \prod_1^4 z_i$$

Les relations qui lient les coefficients aux racines sont :

$$\frac{1}{A_1} = S_1$$

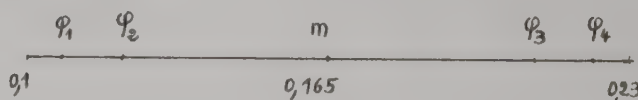
$$B_1 + D_1 = \frac{1}{2} (S_1^2 - S_2)$$

$$\frac{D_1}{A_1} = P S_{-1}$$

$$B_1 D_1 - C_1 = P$$

Ces relations permettent de déterminer A_1, B_1, C_1, D_1 à partir des valeurs $\varphi_1, \varphi_2, \varphi_3, \varphi_4$.

Pour simplifier, nous choisissons $\varphi_1, \varphi_2, \varphi_3, \varphi_4$ symétriques par rapport au milieu m de l'intervalle l_1 . Ceci permet de faire le changement de variable $a_1 = m + t$ ($|t| \leq 0,065$) et par suite de n'avoir plus à déterminer que deux valeurs α et β au lieu de quatre.



$$\varphi_1 = m - \alpha ; \quad \varphi_2 = m - \beta ; \quad \varphi_3 = m + \beta ; \quad \varphi_4 = m + \alpha .$$

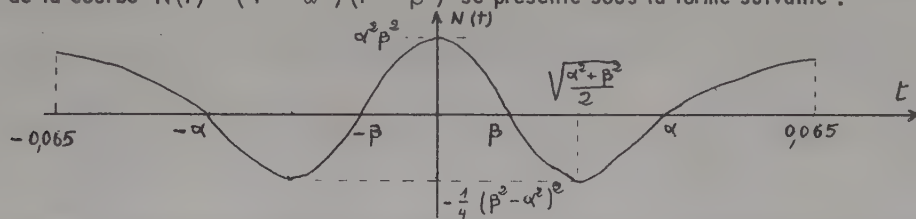
L'écart $(x_0 - \sqrt{a_1})$, qui s'annule pour $\varphi_1, \varphi_2, \varphi_3, \varphi_4$, peut s'écrire :

$$\frac{(a_1 - \varphi_1)(a_1 - \varphi_2)(a_1 - \varphi_3)(a_1 - \varphi_4)}{\frac{1}{A_1}(D_1 + a_1)} = \frac{(t^2 - \alpha^2)(t^2 - \beta^2)}{\frac{D(t)}{A_1}}$$

Le dénominateur $\frac{D_1 + m + t}{A_1}$ varie relativement peu pour $|t| \leq 0,065$.

$$\text{d'où } (t^2 - \alpha^2)(t^2 - \beta^2) \simeq K(x_0 - \sqrt{a_1}) \text{ avec } K = \frac{D_1 + m}{A_1}$$

Le tracé de la courbe $N(t) = (t^2 - \alpha^2)(t^2 - \beta^2)$ se présente sous la forme suivante :



Puisqu'il s'agit d'une courbe d'erreur, nous choisissons α et β de telle sorte que cette courbe oscille également de part et d'autre de l'axe des abscisses, sur tout l'intervalle $(-0,065 ; 0,065)$ y compris les bornes. Il suffit donc que les extrêma soient égaux en valeur absolue :

$$\alpha^2 \beta^2 = \frac{1}{4}(\beta^2 - \alpha^2)^2 = (\overline{0,065^2} - \alpha^2)(\overline{0,065^2} - \beta^2)$$

Ces deux équations donnent :

$$\alpha = 0,0600535$$

$$\beta = 0,0248739$$

Nous retenons :

$$\alpha = 0,06$$

$$\beta = 0,025$$

Toutes les autres données sont à présent calculables :

$$\varphi_1 = 0,105$$

$$z_1 = 0,324\,037\,04$$

$$S_1 = 1,608\,434\,32$$

$$\varphi_2 = 0,14$$

$$z_2 = 0,374\,165\,74$$

$$S_2 = 0,660$$

$$\varphi_3 = 0,19$$

$$z_3 = 0,435\,889\,89$$

$$S_{-1} = 10,161\,021\,825$$

$$\varphi_4 = 0,225$$

$$z_4 = 0,474\,341\,65$$

$$P = 0,025\,068\,407$$

$$\mu = \sqrt{2,3} = 1,516\,575\,09$$

$I_1 (0,1 ; 0,23)$	$I_2 (0,23 ; 0,529)$	$I_3 (0,529 ; 1,2167)$
$A_1 = 0,621722620$	$A_2 = 0,409951755$	$A_3 = 0,270314182$
$B_1 = 0,805164872$	$B_2 = 1,851879206$	$B_3 = 4,259322172$
$C_1 = 0,102441995$	$C_2 = 0,541918154$	$C_3 = 2,866747032$
$D_1 = 0,158365577$	$D_2 = 0,364240827$	$D_3 = 0,837753902$

L'écart maximum $(x_0 - \sqrt{a_1})$ est : $\frac{A_1 \alpha^2 \beta^2}{D_1 + m}$

Soit $\frac{(0,622)(0,06)^2(0,025)^2}{(0,323)} < 4,4 \cdot 10^{-6}$

Dans ce cas, une seule application de la formule de Newton suffit pour obtenir la racine avec un écart de 1 sur le 10^e chiffre.

En effet de : $(x_1 - \sqrt{a}) < \frac{B}{2|f'(x_0)|} (x_0 - \sqrt{a})^2$

Nous avons :

$$(x_1 - \sqrt{a}) < \frac{1}{0,2} (4,4 \cdot 10^{-6})^2 < 1 \cdot 10^{-10}$$

112 - Comparaison suivant le choix de x_0

Nous donnons dans un tableau le nombre d'opérations à effectuer suivant que nous utilisons pour valeur de départ $x_0 = \frac{1}{2} \left(\frac{a}{k} + k \right)$ ou $x_0 = A \left(B + a - \frac{C}{D+a} \right)$

Dans ce tableau « n » représente le nombre d'itérations. A chaque itération, il y a une addition supplémentaire pour former $x_{n+1} - x_n$ qui est nécessaire au test d'arrêt.

Nature des Opérations	1 ^{er} procédé $x_0 = \frac{1}{2} \left(\frac{a}{k} + k \right)$	2 ^{ème} procédé $x_0 = A \left(B + a - \frac{C}{D+a} \right)$
Additions	$2n+1$	4
Multiplications	$n+1$	2
Divisions	$n+1$	2

Dans le premier procédé, le nombre d'opérations dépend du nombre d'itérations : le temps de calcul est donc variable. Dans le second, au contraire, puisqu'une seule application de la formule de Newton suffit, quelle que soit la valeur de l'argument, le nombre d'opérations est constant et, par suite, le temps de calcul aussi. Le tableau nous montre que ce temps est plus court puisque dans l'ensemble des cas « n » est supérieur à 1.

Dans le premier procédé, l'expression de x_0 et la formule de Newton se présentent sous des formes

identiques : nous pouvons alors utiliser la même séquence d'instructions pour les calculs de x_0 et de x_{n+1} . Il en résulte un encombrement restreint. Il n'en est pas de même dans le second où l'expression de x_0 et la formule de Newton ont des formes différentes et ne permettent donc pas d'exécuter les calculs par la même séquence. De plus, nous devons introduire les instructions qui permettent de localiser l'argument dans l'un des trois sous-intervalles I_1, I_2, I_3 ainsi que les douze coefficients nécessaires au calcul de x_0 . L'encombrement est donc beaucoup plus important avec ce procédé.

En résumé, le premier procédé requiert peu d'instructions mais un temps de calcul long et variable ; le second présente les caractéristiques inverses.

12 - FONCTIONS TRANSCENDANTES

121 - Etude théorique des différentes approximations

121 - 1 Approximations polynomiales

121 - 11 Développement en polynômes de Tchebycheff

121 - 111 . Principe .

Sous réserve qu'elle vérifie certaines conditions, une fonction de $\cos \theta$ ($0 \leq \theta \leq \pi$) peut être développée en la série trigonométrique suivante :

$$f(\cos \theta) = \frac{a_0}{2} + \sum_{h=1}^{\infty} a_h \cos h \theta$$

où a_h s'exprime par l'intégrale définie :

$$a_h = \frac{2}{\pi} \int_0^{\pi} f(\cos \theta) \cos h \theta d\theta$$

Si nous posons $T_h(X) = \cos h \theta$ avec $X = \cos \theta$, nous obtenons par définition les « polynômes de Tchebycheff ».

Dans la suite, si nous ramenons l'intervalle de variation de X à $[-1 ; +1]$ nous pouvons remplacer dans le développement de $f(\cos \theta)$, $\cos \theta$ par X et $\cos h \theta$ par $T_h(X)$: nous obtenons ainsi le développement de la fonction $f(X)$ suivant les polynômes de Tchebycheff .

$$f(X) = \frac{a_0}{2} + \sum_{h=1}^{\infty} a_h T_h(X)$$

avec

$$a_h = \frac{2}{\pi} \int_{-1}^{+1} f(X) \cdot T_h(X) \frac{dX}{\sqrt{1-X^2}}$$

REMARQUE - Si nous supposons la fonction $f(x)$ continue sur l'intervalle $[\alpha, \beta]$ ce dernier est ramené à $[-1, 1]$ par le changement de variable :

$$X = \frac{2x - \alpha - \beta}{\beta - \alpha}$$

Calcul des a_h

Nous admettons que la fonction $f(X)$ est analytique sur l'intervalle $[-1, +1]$

$$f(X) = \sum_{q=0}^{\infty} \frac{X^q}{q!} f^{(q)}(0)$$

Nous cherchons le développement de X^q suivant les polynômes de Tchebycheff.

1) $q = 2p$

$$X^{2p} = (\cos \theta)^{2p} = \frac{(e^{i\theta} + e^{-i\theta})^{2p}}{2^{2p}} = \frac{1}{2^{2p}} \left[\sum_{m=0}^{2p} \binom{2p}{m} e^{2(p-m)i\theta} \right]$$

avec $\binom{2p}{m} = \frac{2p!}{m! (2p-m)!}$

La quantité entre crochets peut être fractionnée en :

$$\sum_{m=0}^{p-1} \binom{2p}{m} e^{2(p-m)i\theta} + \binom{2p}{p} + \sum_{m=p+1}^{2p} \binom{2p}{m} e^{2(p-m)i\theta}$$

La somme étant commutative, nous pouvons intervertir les limites de variation de m

$$\sum_{m=p+1}^{2p} = \sum_{m=2p}^{p+1} = \binom{2p}{2p} e^{-2pi\theta} + \binom{2p}{2p-1} e^{-2(p-1)i\theta} + \dots + \binom{2p}{p+1} e^{-2i\theta}$$

$$\sum_{m=0}^{p-1} \binom{2p}{m} e^{2(p-m)i\theta} = \binom{2p}{0} e^{2pi\theta} + \binom{2p}{1} e^{2(p-1)i\theta} + \dots + \binom{2p}{p-1} e^{2i\theta}$$

Or :

$$\binom{2p}{m} = \frac{2p!}{m! (2p-m)!} = \binom{2p}{2p-m}$$

La somme des deux quantités précédentes devient :

$$\sum_{m=0}^{p-1} \binom{2p}{m} (e^{2(p-m)i\theta} + e^{-2(p-m)i\theta}) = 2 \sum_{m=0}^{p-1} \binom{2p}{m} \cos 2(p-m)\theta$$

En posant $p-m = n$ et $\cos 2n\theta = T_{2n}(X)$

$$2^{2p} X^{2p} = \binom{2p}{p} + 2 \sum_{n=1}^p \binom{2p}{p-n} T_{2n}(X)$$

2) $q = 2p+1$

La même démonstration nous donne :

$$2^{2p+1} X^{2p+1} = 2 \sum_{n=0}^p \binom{2p+1}{p-n} T_{2n+1}(X)$$

D'après la formule de définition de $T_n(X)$ nous avons :

$$T_n(X) = T_{-n}(X) \quad \text{et} \quad T_0(X) = 1$$

d'où les deux expressions :

$$2^{2p} X^{2p} = \sum_{n=1}^p \binom{2p}{p-n} T_{2n}(X) + \binom{2p}{p} T_0(X) + \sum_{n=1}^p \binom{2p}{p-n} T_{-2n}(X)$$

$$2^{2p+1} X^{2p+1} = \sum_{n=0}^p \binom{2p+1}{p-n} T_{2n+1}(X) + \sum_{n=0}^p \binom{2p+1}{p-n} T_{-(2n+1)}(X)$$

qui peuvent se regrouper en une seule, en posant $2p$ ou $2p+1 = q$ et en ordonnant le développement suivant les ordres décroissants des $T(X)$:

$$2^q X^q = \sum_{i=0}^q \binom{q}{i} T_{q-2i}(X)$$

Le coefficient a_h devient :

$$a_h = \frac{2}{\pi} \int_{-1}^{+1} \left\{ \sum_{q=0}^{\infty} \frac{f^{(q)}(0)}{2^q q!} \left[\sum_{i=0}^q \binom{q}{i} T_{q-2i}(X) \right] \right\} \frac{T_h(X) dX}{\sqrt{1-X^2}}$$

L'expression de a_h peut être intégrée terme à terme. En effet la série $\sum U_q(X)$ sous le signe $\int$ converge uniformément pour $-1 \leq X \leq +1$.

Si nous reprenons l'ancienne variable θ , nous obtenons la série de terme général :

$$U_q(\theta) = \frac{f^{(q)}\left(\frac{\pi}{2}\right) (\cos \theta)^q \cos h\theta}{q!}$$

Pour $0 \leq \theta \leq \pi$, le module de chaque terme $U_q(\theta)$ est majoré par le nombre $V_q = \frac{f^{(q)}\left(\frac{\pi}{2}\right)}{q!}$

La série de terme général V_q converge puisqu'elle admet pour somme $f(1)$ que nous obtenons en faisant $X = \cos \theta = 1$ dans $\sum_{q=0}^{\infty} \frac{X^q}{q!} f^{(q)}(0) = f(X)$

$\sum_{q=0}^{\infty} U_q(\theta)$ converge donc normalement pour $0 \leq \theta \leq \pi$, donc uniformément.

Si nous reprenons la variable X , $\sum_{q=0}^{\infty} U_q(X)$ reste uniformément convergent. Ainsi, chaque terme du développement de a_h se présente sous la forme :

$$K \int_{-1}^{+1} T_p(X) \cdot T_h(X) \frac{dX}{\sqrt{1-X^2}}$$

($p = 0, 1 \dots h-1, h, h+1, \dots$)

Mais les polynômes de Tchebycheff possèdent la propriété d'orthogonalité dans l'intervalle $[-1, 1]$ par rapport à $\frac{1}{\sqrt{1-X^2}}$. Aussi, dans le développement de a_h , seuls les termes qui se présentent sous la forme d'un produit de polynômes de Tchebycheff du même ordre, sont conservés; les autres sont nuls.

$$0 \quad \text{si } h \neq p > 0$$

$$\int_{-1}^{+1} T_p(X) \cdot T_h(X) \frac{dX}{\sqrt{1-X^2}} = \frac{\pi}{2} \quad \text{si } h=p > 0$$

$$\pi \quad \text{si } h = p = 0$$

d'où :

$$a_h = \frac{1}{2^{h-1} h!} \left[f^{(h)}(0) + \sum_{k=1}^{\infty} \frac{f^{(h+2k)}(0)}{2^{2k} k! (h+1) \dots (h+k)} \right]$$

De :

$$\cos(h+1)\theta + \cos(h-1)\theta = 2 \cos \theta \cos h \theta, \text{ nous déduisons la formule de}$$

réurrence :

$$T_{h+1}(X) = 2X T_h(X) - T_{h-1}(X)$$

avec

$$T_0(X) = 1 \quad \text{et} \quad T_1(X) = X$$

Le développement de la fonction : $f(X) = \frac{a_0}{2} + \sum_{h=1}^{\infty} a_h T_h(X)$ est alors parfaitement déterminé.

121 - 112 . Développement limité à l'ordre (n-1) Erreur de méthode

Le développement de la fonction $f(X)$ limité à l'ordre (n-1) donne une approximation de celle-ci. Le choix de l'écart désiré entre $f(X)$ et son approximation permet de déterminer « n » et l'erreur de méthode correspondante.

Si nous voulons $| \text{fonction} - \text{approximation} | \leq \varepsilon$

$$\text{soit } \left| \frac{a_0}{2} + \sum_{h=1}^{\infty} a_h T_h(X) - \frac{a_0}{2} - \sum_{h=1}^{n-1} a_h T_h(X) \right| \leq \varepsilon$$

$$\text{ou } \left| \sum_{h=n}^{\infty} a_h T_h(X) \right| \leq \varepsilon$$

chaque polynôme de Tchebycheff étant borné en module par 1, une condition suffisante est :

$$|a_n| + |a_{n+1}| + \dots \leq \varepsilon$$

1) Tous les polynômes de Tchebycheff sont d'ordre pair ou impair.

L'expérience montre que, pour les fonctions étudiées, les coefficients décroissent alors très rapidement. En effet, nous avons :

$$a_k = \lambda a_{k+2}$$

avec λ de l'ordre de 10^{-2} dans les cas les plus défavorables.

La condition suffisante devient :

$$|a_n| [1 + |\lambda| + \lambda^2 + \dots] = \frac{|a_n|}{1 - |\lambda|} \simeq |a_n| \leq \varepsilon$$

Il suffit de choisir « n » de façon que $|a_n| < \varepsilon$. L'erreur de méthode est alors de l'ordre de $|a_n|$

2) Les polynômes de Tchebycheff sont d'ordre consécutif.

Dans ce cas, nous vérifions toujours pour les fonctions étudiées que

$$|a_k| \cdot |a_{k+2}| < |a_{k+1}|^2 \text{ quel que soit } k.$$

La série $|a_n| + |a_{n+1}| + \dots + |a_{n+p}| + \dots$ peut se mettre sous la forme :

$$|a_n| \left[1 + \frac{|a_{n+1}|}{|a_n|} + \frac{|a_{n+2}|}{|a_{n+1}|} \cdot \frac{|a_{n+1}|}{|a_n|} + \dots + \underbrace{\frac{|a_{n+p}|}{|a_{n+p-1}|} \cdot \frac{|a_{n+p-1}|}{|a_{n+p-2}|} \cdot \dots \cdot \frac{|a_{n+1}|}{|a_n|}}_{p \text{ produits}} + \dots \right]$$

Or :

$$\frac{|a_{n+p}|}{|a_{n+p-1}|} < \frac{|a_{n+p-1}|}{|a_{n+p-2}|} < \dots < \frac{|a_{n+1}|}{|a_n|}$$

La quantité précédente est donc majorée par :

$$|a_n| \left[1 + \frac{|a_{n+1}|}{|a_n|} + \dots + \left(\frac{|a_{n+1}|}{|a_n|} \right)^p + \dots \right] = |a_n| \left[\frac{1}{1 - \frac{|a_{n+1}|}{|a_n|}} \right]$$

Il suffit de choisir n de façon que :

$$|a_n| \frac{1}{1 - \frac{|a_{n+1}|}{|a_n|}} \leq \varepsilon$$

L'erreur de méthode admet alors pour borne supérieure :

$$\frac{|a_n|}{1 - \frac{|a_{n+1}|}{|a_n|}}$$

121 - 113 - Développement suivant les puissances de X

Dans la suite, nous désignerons l'approximation de la fonction $f(X)$ par $A_p[f(X)]$

$$A_p[f(X)] = \frac{a_0}{2} + \sum_{h=1}^{n+1} a_h T_h(X)$$

chaque polynôme $T_h(X)$ peut s'exprimer suivant les puissances décroissantes de X .

Des relations suivantes :

$$T_h(X) = \cos h \theta = \sum_{i=0}^h \binom{h}{2i} \cos^{h-2i} \theta \cdot \sin^{2i} \theta$$

$$\sin^{2i} \theta = \sum_{j=0}^i (-1)^j \binom{i}{j} \cos^{2i-2j} \theta$$

nous déduisons :

$$T_h(X) = 2^{h-1} \left[X^h - \frac{h}{1! 2^2} X^{h-2} + \frac{h(h-3)}{2! 2^4} X^{h-4} - \frac{h(h-4)(h-5)}{3! 2^6} X^{h-6} + \dots \right]$$

Le dernier terme de l'expression comprise entre crochets est égal à :

$$(-1)^{\frac{h}{2}} \frac{1}{2^{h-1}} \quad \text{si } h \text{ est pair.}$$

$$(-1)^{\frac{h-1}{2}} \frac{hX}{2^{h-1}} \quad \text{si } h \text{ est impair.}$$

Il suffit de regrouper les termes de même puissance en X pour obtenir une approximation de la fonction ordonnée en X .

121 - 12 - Développement en série de puissances et Relaxation.

Les polynômes de Tchebycheff permettent décomposer un ou plusieurs termes dans les développements en série de puissances limités à une somme partielle :

$$\text{De} \quad T_h(X) = 2^{h-1} \left[X^h - \frac{h}{1! 2^2} X^{h-2} + \frac{h(h-3)}{2! 2^4} X^{h-4} - \dots \right]$$

nous tirons :

$$X^h - P_{h-2}(X) = \frac{T_h(X)}{2^{h-1}}$$

L'erreur provoquée par le remplacement de X^h par un polynôme d'ordre $h-2$ en X est inférieure ou égale en valeur absolue à :

$$\left| \frac{T_h(X)}{2^{h-1}} \right| \leq \frac{1}{2^{h-1}} \quad (|T_h(X)| \leq 1)$$

Le terme $a_h X^h$ est dit « relaxé » s'il est remplacé par $a_h P_{h-2}(X)$

Soit à calculer $f(X) = \sum_{h=0}^{\infty} a_h X^h$ ($|X| \leq 1$) avec une précision telle que :

$$|\text{fonction} - \text{approximation}| \leq \varepsilon$$

Une approximation de $f(X)$ est :

$$\text{Ap. } [f(X)] = a_0 + a_1 X + \dots + a_{n-1} X^{n-1}$$

Il suffit de choisir l'ordre « n » de façon que l'inégalité $\left| \sum_{h=n}^{\infty} a_h X^h \right| \leq \varepsilon$ soit vérifiée.

Le terme $a_{n-1} X^{n-1}$ peut être relaxé c'est-à-dire remplacé par $a_{n-1} P_{n-3}(X)$, à condition de vérifier :

$$\left| \frac{a_{n-1}}{2^{n-2}} \right| \leq \varepsilon \rightarrow \left| \sum_{h=n}^{\infty} a_h X^h \right|$$

En effet dans les cas défavorables, il peut y avoir cumul des erreurs qui résultent de la troncature et de la relaxation. L'inégalité précédente suffit à assurer que la somme des erreurs reste bien inférieure ou, à la rigueur, égale, à ε .

Après la correction des coefficients d'ordre inférieur, nous obtenons une approximation d'ordre $h-2$ (au lieu de $h-1$) qui vérifie toujours :

$$| \text{fonction} - \text{Approximation} | \leq \varepsilon \quad . \text{ Si le coefficient corrigé } a_{n+2}^1 \text{ vérifie :}$$

$$\left| \frac{a_{n+2}^1}{2^{n+3}} \right| \leq \varepsilon - \left| \frac{a_{n+1}^1}{2^{n+2}} \right| - \left| \sum_{h=n}^{\infty} a_h^1 X^h \right|$$

Le terme $a_{n+2}^1 X^{n+2}$ peut être à son tour relaxé .

121 - 13 - Développement d'Hastings .

Nous retranscrivons, lors de l'étude de chaque fonction, les développements limités que donne Hastings dans « Approximations for Digital Computers » .

Notons tout de suite que l'erreur de méthode obtenue par ces développements est plus importante que celle que nous avons fixée dans notre étude, ce qui rend la comparaison difficile.

121 - 2 Approximation rationnelle

La fonction à calculer est mise sous la forme : $f(x) = R(x) + C(x)$.

$R(x)$ est une fraction rationnelle en x qui constitue l'approximation de la fonction $f(x)$; $C(x)$ est un terme complémentaire tel que $|C(x)| < \varepsilon$ est vérifié sur tout l'intervalle de variation de x .

Pour obtenir $R(x)$ et $C(x)$, les méthodes sont différentes suivant les fonctions étudiées ; c'est pourquoi nous les développerons succinctement lors de l'étude particulière à chaque fonction.

$$\begin{array}{c} \div \\ \div \quad \div \\ \div \end{array}$$

122 - Application aux fonctions transcendentes

Les fonctions traitées dans notre étude sont : Logarithme népérien, exponentielle, sinus, cosinus, Arc tangente et Arc sinus .

Pour effectuer les calculs propres à chacune d'elles, il faut préalablement fixer l'écart maximum ε entre la fonction et son approximation . Nous choisissons :

$$\varepsilon = 0,5 \cdot 10^{-9} \text{ en simple précision}$$

$$\varepsilon = 0,5 \cdot 10^{-18} \text{ en double précision .}$$

Les raisons de ce choix seront exposées dans la seconde partie . (Voir paragraphe 212) .

122- 1 - Logarithme Népéri en

122-11 - Calculs numériques des approximations .

Quelle que soit l'approximation étudiée, son calcul se fera toujours au voisinage de l'unité afin d'obtenir pour l'approximation un nombre de termes restreint. En premier lieu, il convient de réduire l'intervalle de variation de l'argument en un intervalle aussi étroit que possible et comprenant l'unité . Pour cette dernière raison, nous choisissons comme intervalle de réduction $[10^{-\alpha}, 10^{\alpha}]$ avec $\alpha = \frac{1}{2^q}$. Le nombre de termes de l'approximation est d'autant plus faible que q est grand. Nous nous limitons toutefois à $q = 4$.

122 - 111 - Approximations polynomiales .

122-111-1 - Développement de Tchebycheff .

122-111-11 Développement de Tchebycheff de Log (1 + λ X)

Si x est l'argument réduit dans l'intervalle $[10^{-1/16}, 10^{1/16}]$, nous effectuons le changement de variable :

$$x = 1 + \lambda X$$

λ est un facteur numérique à déterminer de façon que l'intervalle de variation de X soit $[-1, +1]$.

Si nous prenons 1,16 pour valeur approchée par excès de $10^{1/16}$, nous lui faisons correspondre $X=1$ et en déduisons $\lambda = 0,16$.

$$\text{Soit : } x = 1 + 0,16 X \text{ avec } |X| \leq 1$$

Le développement de Tchebycheff est :

$$\text{Log}(1 + \lambda X) = \frac{a_0}{2} + \sum_{h=1}^{\infty} a_h T_h(X)$$

Calcul des coefficients a_h

Si $f(x) = \text{Log } x$, les dérivées successives pour $X = 0$ sont de la forme :

$$f_{(1)}^{(h)} = (-1)^{h+1} (h-1)! \lambda^h$$

d'où , en reportant l'expression de $f_{(1)}^{(h)}$ dans a_h :

$$a_h = \frac{(-1)^{h+1} \lambda^h}{2^{h-1}} \left[\frac{1}{h} + \sum_{k=1}^{\infty} \frac{(h+2k-1)!}{2^{2k} k! (h+k)!} \lambda^{2k} \right]$$

Les coefficients a_h sont calculables dès à présent .

Posons :

$$U_k = \frac{(h+2k-1)!}{2^{2k} k! (h+k)!} \lambda^{2k}$$

Nous avons :

$$\frac{U_{k+1}}{U_k} = \frac{(h+2k)(h+2k+1)}{4(k+1)(h+k+1)} \lambda^2$$

Cette formule de récurrence permet de passer de U_k à U_{k+1} connaissant $U_1 = \frac{\lambda^2}{4}$.

Pour $h=0$, $f_{(1)}^{(0)} = 0$ d'où $a_0 = -2 \sum_{k=1}^{\infty} U_k$

Ce procédé de calcul présente l'inconvénient d'être long car la série $\sum_{k=0}^{\infty} U_k$ a une faible convergence. Or, comme les coefficients a_h seront également utilisés en double précision, il faut effectuer leur calcul en triple précision.

Mais puisque la série $\sum_{k=0}^{\infty} U_k$ converge (sa limite étant $(-1)^{h+1} \frac{2^{h-1}}{\lambda^h} a_h - \frac{1}{h}$), il est préférable, pour réduire la longueur des calculs, de chercher théoriquement sa limite quand k tend vers l'infini.

a_h peut s'écrire sous la forme plus condensée :

$$a_h = (-1)^{h+1} 2 \sum_{k=0}^{\infty} \frac{(h+2k-1)!}{k! (h+k)!} \left(\frac{\lambda}{2} \right)^{h+2k}$$

Nous admettons que $\frac{\lambda}{2}$ peut se mettre sous la forme : $\frac{t}{1+t^2}$ t étant une fonction de λ .

Le terme général de la série devient :

$$\frac{(h+2k-1)!}{k! (h+k)!} t^{h+2k} (1+t^2)^{-(h+2k)}$$

compte tenu de la valeur de λ , nous en déduisons que t^2 est inférieur à 1 et que par suite, nous pouvons développer $(1+t^2)^{-(h+2k)}$ suivant :

$$1 - (h+2k)t^2 + \frac{(h+2k)(h+2k+1)}{2!} t^4 - \dots (-1)^r \frac{(h+2k+r-1)!}{(h+2k-1)! r!} t^r \dots$$

Dans le développement de $\sum_{k=0}^{\infty} \frac{(h+2k-1)!}{k! (h+k)!} t^{h+2k} [1 - (h+2k)t^2 + \dots]$ nous regroupons les termes de même puissance en t .

Nous obtenons :

$$\frac{(h-1)!}{h!} t^h - \left[\frac{(h-1)! h}{h!} - \frac{(h+1)!}{1! (h+1)!} \right] t^{h+2} + \left[\frac{(h-1)!}{h!} \frac{h(h+1)}{2!} - \frac{(h+1)!}{1! (h+1)!} (h+2) + \frac{(h+3)!}{2! (h+2)!} \right] t^{h+4} - \dots$$

Le calcul des coefficients de t^{h+2k} donne un résultat nul dès que k est supérieur ou égal à 1.

La limite de la série est donc égale à $\frac{t^h}{h}$.

Si nous posons $\lambda = \sin 2\alpha$, la relation $\frac{\lambda}{2} = \frac{t}{1+t^2}$ donne $t = \tan \alpha$.

La valeur de a_h peut s'exprimer très simplement par :

$$a_h = (-1)^{h+1} 2 \frac{(\tan \alpha)^h}{h} \quad \text{avec } \sin 2\alpha = \lambda$$

Le calcul de a_0 s'obtient en faisant $X = 0$ dans l'expression : $\text{Log}(1+\lambda X) = \frac{a_0}{2} + \sum_{h=1}^{\infty} a_h T_h(X)$

Nous en déduisons :

$$-\frac{a_0}{2} = \sum_{h=1}^{\infty} a_h T_h(0)$$

Par définition, à $X = \cos \theta = 0$ correspond $\theta = \frac{\pi}{2}$ soit $T_h(0) = \cos h \frac{\pi}{2}$ dont les valeurs sont :

$$T_h(0) = 0 \text{ si } h = 2p+1$$

$$T_h(0) = -1 \text{ si } h = 2(2p+1)$$

$$T_h(0) = 1 \text{ si } h = 4p$$

Il en résulte :

$$-\frac{a_0}{2} = \left[-a_2 + a_4 - a_6 + a_8 - \dots \right] \\ 2 \left[\frac{\tan^2 \alpha}{2} - \frac{\tan^4 \alpha}{4} + \frac{\tan^6 \alpha}{6} - \dots \right]$$

Après simplification par 2, nous retrouvons le développement de $\text{Log}(1+\text{tg}^2 \alpha)$

$$\text{d'où : } -\frac{a_0}{2} = \text{Log}(1+\text{tg}^2 \alpha) = \text{Log} \frac{1}{\cos^2 \alpha} = -2 \text{Log}(\cos \alpha)$$

$$\boxed{a_0 = 4 \text{Log}(\cos \alpha)} \quad \text{avec } \sin 2\alpha = \lambda$$

Valeurs des 20 premiers coefficients a_h

Les coefficients ont été calculés en triple précision. Le nombre entre parenthèses représente la puissance de 10 par laquelle il faut multiplier la valeur donnée.

$\frac{a_0}{2} =$	-	6,462	328	769	381	801	521	249	290	72	(-3)
$a_1 =$		1,610	373	207	469	340	364	581	567	26	(-1)
$a_2 =$	-	6,483	254	668	337	727	863	479	539	37	(-3)
$a_3 =$		3,480	153	205	030	533	762	772	527	05	(-4)
$a_4 =$	-	2,101	629	554	726	147	085	947	442	05	(-5)
$a_5 =$		1,353	763	170	782	682	828	392	581	45	(-6)
$a_6 =$	-	9,083	599	747	863	222	277	310	978	79	(-8)
$a_7 =$		6,269	136	712	000	366	574	332	406	35	(-9)
$a_8 =$	-	4,416	846	785	298	423	269	283	534	19	(-10)
$a_9 =$		3,161	231	877	574	074	138	983	304	39	(-11)
$a_{10} =$	-	2,290	843	403	209	479	092	939	008	19	(-12)
$a_{11} =$		1,676	869	472	289	285	553	526	025	86	(-13)
$a_{12} =$	-	1,237	676	765	690	711	929	939	780	42	(-14)
$a_{13} =$		9,199	022	321	425	989	254	647	833	46	(-16)
$a_{14} =$	-	6,877	863	144	906	384	251	811	737	98	(-17)
$a_{15} =$		5,168	765	715	492	427	308	557	197	24	(-18)
$a_{16} =$	-	3,901	707	104	960	203	188	019	478	19	(-19)
$a_{17} =$		2,956	802	157	750	906	659	009	441	96	(-20)
$a_{18} =$	-	2,248	512	071	352	863	864	661	058	84	(-21)
$a_{19} =$		1,715	183	808	810	651	324	806	898	03	(-22)

Développement limité à l'ordre $(n-1)$ - Erreur de méthode

Il suffit de choisir « n » de façon que l'inégalité suivante soit vérifiée.

$$\frac{|a_n|}{1 - \frac{|a_{n+1}|}{|a_n|}} \leq \varepsilon$$

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

Le tableau précédent montre que :

$$\frac{|a_8|}{1 - \frac{|a_9|}{|a_8|}} < 4,8 \cdot 10^{-10} < \varepsilon$$

Le développement limité à l'ordre 7 fournit une approximation de $\text{Log}(1+\lambda X)$:

$$Ap. [\text{Log}(1+\lambda X)] = \frac{a_0}{2} + \sum_{h=1}^7 a_h T_h(X)$$

L'erreur de méthode correspondante est inférieure à 5.10^{-10} .

2) Double précision ($\epsilon = 0.5.10^{-18}$)

$$\frac{|a_{16}|}{1 - \frac{|a_{17}|}{|a_{16}|}} < 4.25.10^{-19} < \epsilon$$

une approximation de $\text{Log}(1+\lambda X)$ est :

$$Ap. [\text{Log}(1+\lambda X)] = \frac{a_0}{2} + \sum_{h=1}^{16} a_h T_h(X)$$

L'erreur de méthode correspondante est inférieure à 5.10^{-19} .

Si nous remplaçons $T_h(X)$ par son expression en X , nous obtenons des approximations de $\text{Log}(1+\lambda X)$ développées suivant les puissances croissantes de X

Formules numériques

1) Simple précision

$$Ap. [\text{Log}(1+0.16 X)] = \sum_{n=0}^7 K_n X^n \text{ avec } |X| \leq 1$$

$$K_0 = 0,000\,000\,000\,44$$

$$K_4 = -0,000\,163\,770\,08$$

$$K_1 = 0,159\,999\,999\,70$$

$$K_5 = 0,000\,020\,957\,92$$

$$K_2 = -0,012\,800\,014\,06$$

$$K_6 = -0,000\,002\,906\,88$$

$$K_3 = 0,001\,365\,337\,20$$

$$K_7 = 0,000\,000\,401\,28$$

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 5.10^{-10}$$

2) Double précision

$$Ap. [\text{Log}(1+0.16 X)] = \sum_{n=0}^{16} K_n X^n \text{ avec } |X| \leq 1$$

$$K_0 = 0,000\,000\,000\,000\,000\,000\,38$$

$$K_1 = 0,159\,999\,999\,999\,999\,999\,44$$

$$K_2 = -0,012\,800\,000\,000\,000\,000\,34$$

$$K_3 = 0,001\,365\,333\,333\,333\,358\,92$$

$$K_4 = -0,000\,163\,839\,999\,998\,962\,08$$

$$K_5 = 0,000\,020\,971\,519\,999\,650\,08$$

$$K_6 = -0,000\,002\,796\,202\,674\,959\,04$$

$$K_7 = 0,000\,000\,383\,479\,225\,016\,32$$

$$K_8 = -0,000\,000\,053\,687\,058\,727\,68$$

$$K_9 = 0,000\,000\,007\,635\,490\,324\,48$$

$$K_{10} = -0,000\,000\,001\,099\,580\,528\,64$$

$$K_{11} = 0,000\,000\,000\,159\,942\,195\,20$$

$$K_{12} = -0,000\,000\,000\,023\,375\,564\,80$$

$$K_{13} = 0,000\,000\,000\,003\,450\,265\,60$$

$$K_{14} = -0,000\,000\,000\,000\,563\,445\,76$$

$$K_{15} = 0,000\,000\,000\,000\,084\,705\,28$$

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 5.10^{-19}$$

122-111-12 Développement de Tchebycheff de $\text{Log} \frac{1+\lambda X}{1-\lambda X}$

Il est préférable de développer $\text{Log} \frac{1+\lambda X}{1-\lambda X}$, qui converge plus rapidement que

$\text{Log}(1+\lambda X)$.

Si x est l'argument réduit dans l'intervalle $[10^{-1/10}, 10^{1/10}]$, nous effectuons le changement de variable :

$$x = \frac{1+\lambda X}{1-\lambda X}$$

λ est un facteur numérique à déterminer de telle sorte que l'intervalle de variation de X soit $[-1, +1]$. Il suffit de faire correspondre les bornes de l'intervalle de variation de x et celles de l'intervalle de X :

$$\frac{1+\lambda}{1-\lambda} = 10^{1/10} \quad \text{soit} \quad \lambda = \frac{10^{1/10} - 1}{10^{1/10} + 1} < 0,072$$

Soit $x = \frac{1+0,072 X}{1-0,072 X}$ avec $|X| \leq 1$

Le développement de Tchebycheff est :

$$\text{Log} \frac{1+\lambda X}{1-\lambda X} = \sum_{h=0}^{\infty} a_{2h+1} T_{2h+1}(X)$$

car $\text{Log} \frac{1+\lambda X}{1-\lambda X}$ étant une fonction impaire, les coefficients d'ordre pair sont nuls.

Calcul des coefficients a_{2h+1}

Nous avons obtenu l'expression de a_{2h+1} par une méthode similaire à celle que nous avons exposée dans le paragraphe 122-111-11.

$$a_{2h+1} = 4 \frac{(\text{tg } \alpha)^{2h+1}}{2h+1} \quad \text{avec} \quad \sin 2\alpha = \lambda$$

Valeurs des 10 premiers coefficients a_{2h+1}

Les coefficients ont été calculés en triple précision. Le nombre entre parenthèses représente la puissance de 10 par laquelle il faut multiplier la valeur donnée.

a_1	=	1,441	871	093	024	008	423	878	382	24	(-1)
a_3	=	6,245	080	888	140	331	882	330	018	67	(-5)
a_5	=	4,868	803	035	069	877	966	650	579	35	(-8)
a_7	=	4,518	840	969	340	421	736	391	706	68	(-11)
a_9	=	4,566	836	628	113	756	953	128	399	96	(-14)
a_{11}	=	4,855	100	089	005	570	107	827	728	86	(-17)
a_{13}	=	5,338	022	595	160	844	075	021	911	65	(-20)
a_{15}	=	6,011	258	283	215	850	864	735	659	15	(-23)
a_{17}	=	6,891	926	126	984	088	816	135	577	36	(-26)
a_{19}	=	8,012	514	373	869	974	390	952	949	86	(-29)

Développement limité à l'ordre $(2n-1)$. Erreur de méthode

Il suffit de choisir « n » de telle sorte que l'inégalité suivante soit vérifiée :

$$|a_{2n+1}| < \varepsilon$$

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

Le tableau précédent montre que : $a_7 < 4,52 \cdot 10^{-11} < \varepsilon$

Le développement limité à l'ordre 5 fournit une approximation de $\text{Log} \frac{1+\lambda X}{1-\lambda X}$

$$\text{Ap.} \left[\text{Log} \frac{1+\lambda X}{1-\lambda X} \right] = \sum_{h=0}^2 a_{2h+1} T_{2h+1}(X)$$

L'erreur de méthode correspondante est inférieure à $5 \cdot 10^{-11}$

2) Double précision ($\varepsilon = 0,5 \cdot 10^{-18}$)

$$a_{13} < 5,34 \cdot 10^{-20} < \varepsilon$$

$$\text{Ap.} \left[\text{Log} \frac{1+\lambda X}{1-\lambda X} \right] = \sum_{h=0}^5 a_{2h+1} T_{2h+1}(X)$$

L'erreur de méthode correspondante est inférieure à $6 \cdot 10^{-20}$

Formules numériques

1) Simple précision

$$\text{Ap.} \left[\text{Log} \frac{1+0,072 X}{1-0,072 X} \right] = \sum_{n=0}^2 K_{2n+1} X^{2n+1} \quad \text{avec } |X| \leq 1$$

$$\begin{aligned} K_1 &= 0,144 \ 000 \ 000 \ 32 \\ K_3 &= 0,000 \ 248 \ 829 \ 47 \\ K_5 &= 0,000 \ 000 \ 779 \ 01 \end{aligned}$$

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 5 \cdot 10^{-11}$$

2) Double précision

$$\text{Ap.} \left[\text{Log} \frac{1+0,072 X}{1-0,072 X} \right] = \sum_{n=0}^5 K_{2n+1} X^{2n+1} \quad \text{avec } |X| \leq 1$$

$$\begin{aligned} K_1 &= 0,143 \ 999 \ 999 \ 999 \ 999 \ 31 \\ K_3 &= 0,000 \ 248 \ 832 \ 000 \ 000 \ 019 \ 40 \\ K_5 &= 0,000 \ 000 \ 773 \ 967 \ 052 \ 644 \ 92 \\ K_7 &= 0,000 \ 000 \ 002 \ 865 \ 889 \ 961 \ 02 \\ K_9 &= 0,000 \ 000 \ 000 \ 011 \ 554 \ 382 \ 15 \\ K_{11} &= 0,000 \ 000 \ 000 \ 000 \ 049 \ 716 \ 22 \end{aligned}$$

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 6 \cdot 10^{-20}$$

122-111-2 Développement en série et Relaxation .

Si x est l'argument réduit dans l'intervalle $[10^{-1/2}, 10^{1/2}]$, nous effectuons le changement de variable :

$$x = \frac{1+t}{1-t}$$

$\frac{dt}{dx} = \frac{2}{(x+1)^2}$ est positif; les valeurs extrêmes de t sont donc atteintes aux bornes de l'intervalle de variation de x . Nous en déduisons :

$$|t| \leq 0,072$$

Le développement en série de Mac-Laurin devient, en fonction de t :

$$\text{Log } x = 2 \sum_{h=0}^{\infty} \frac{t^{2h+1}}{2h+1}$$

Développement limité à l'ordre $(2n-1)$ - Relaxation - Erreur de méthode

L'erreur obtenue, en limitant le développement à l'ordre $(2n-1)$ est :

$$\left| 2 \sum_{h=n}^{\infty} \frac{t^{2h+1}}{2h+1} \right|$$

Cette série est majorée par :

$$\left| \frac{2t^{2n+1}}{2n+1} \right| \sum_{k=0}^{\infty} t^{2k} = \left| \frac{t^{2n+1}}{2n+1} \right| \cdot \frac{2}{1-t^2}$$

Il suffit alors de choisir n tel que :

$$\frac{2(0,072)^{2n+1}}{(2n+1)(1-0,072^2)} \leq \varepsilon$$

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

L'inégalité précédente est vérifiée quand n est supérieur ou égal à 4 .

Le développement limité à l'ordre 7 donne l'approximation

$$\text{Ap. } \left[\text{Log } \frac{1+t}{1-t} \right] = 2 \sum_{h=0}^3 \frac{t^{2h+1}}{2h+1}$$

l'erreur correspondante est inférieure à $1,2 \cdot 10^{-11}$.

Pour pouvoir relaxer le dernier terme de l'approximation, il faut préalablement se ramener à une variable X telle que son intervalle de variation soit $[-1, +1]$ (voir paragraphe 121-12) .

Nous effectuons le changement de variable :

$$t = \lambda X \text{ avec } |X| \leq 1 \text{ et } \lambda = 0,072$$

La relaxation du terme en t^7 donne une erreur de :

$$\frac{2}{7} \lambda^7 \cdot \frac{1}{2^6} \approx 0,45 \cdot 10^{-10} < \varepsilon = 1,2 \cdot 10^{-11}$$

Puisque l'inégalité est vérifiée, nous pouvons remplacer $\frac{2}{7} (\lambda X)^7$ par :

$$\frac{2\lambda^7}{7 \cdot 2^6} [112 X^6 - 56 X^4 + 7X]$$

La relaxation du terme corrigé en t^5 conduit à une erreur supérieure à ε .

L'erreur de méthode est alors inférieure au cumul des erreurs soit 6.10^{-11}

Coefficients corrigés de l'approximation ($\lambda = 0,072$)

$$\text{coefficient de } t : 2 \left(1 + \frac{\lambda^6}{64} \right)$$

$$t^3 : 2 \left(\frac{1}{3} - \frac{\lambda^4}{8} \right)$$

$$t^5 : 2 \left(\frac{1}{5} + \frac{\lambda^2}{4} \right)$$

2) Double précision ($\varepsilon = 0,5.10^{-18}$)

L'inégalité est vérifiée quand n est supérieur ou égal à 8. Le développement limité à l'ordre 15 donne l'approximation :

$$\text{Ap. } \left[\text{Log } \frac{1+t}{1-t} \right] = 2 \sum_{h=0}^7 \frac{t^{2h+1}}{2h+1}$$

L'erreur correspondante est inférieure à $4,4.10^{-21}$

La relaxation du terme en t^{15} donne une erreur de :

$$\frac{2}{15} X^{15} \frac{1}{2^{14}} < 10^{-22} < \varepsilon - 4,4.10^{-21}$$

Nous pouvons remplacer $\frac{2}{15} (\lambda X)^{15}$ par :

$$\frac{2\lambda^{15}}{15.2^{14}} [61440 X^{13} - 92160 X^{11} + 70400 X^9 - 28800 X^7 + 6048 X^5 - 560 X^3 + 15 X]$$

À la suite de cette substitution, le terme en t^{13} devient :

$$\left(\frac{2}{13} + \frac{\lambda^2}{2} \right) t^{13}$$

La relaxation de ce terme donne une erreur de :

$$\left(\frac{2}{13} + \frac{\lambda^2}{2} \right) \lambda^{13} \frac{1}{2^{12}} < 5,4.10^{-20} < \varepsilon - 4,4.10^{-21}$$

$$\left(\frac{2}{13} + \frac{\lambda^2}{2} \right) (\lambda X)^{13} \text{ peut être remplacé par}$$

$$\left(\frac{2}{13} + \frac{\lambda^2}{2} \right) \frac{\lambda^{13}}{2^{12}} [13312 X^{11} - 16640 X^9 + 9984 X^7 - 2912 X^5 + 364 X^3 - 13 X]$$

Le terme corrigé en t^{11} ne peut pas être relaxé.

L'erreur de méthode est alors inférieure au cumul des erreurs soit 6.10^{-20}

Coefficients corrigés de l'approximation ($\lambda = 0,072$)

$$\text{coefficient de } t : 2 - \frac{3\lambda^{14}}{2048} - \frac{\lambda^{12}}{2048}$$

$$t^3 : \frac{2}{3} + \frac{245\lambda^{12}}{6144} + \frac{91\lambda^{10}}{6656}$$

$$t^5 : \frac{2}{5} - \frac{49\lambda^{10}}{160} - \frac{91\lambda^8}{832}$$

$$\text{coefficient de } t^7 : \frac{2}{7} + \frac{63\lambda^6}{64} + \frac{39\lambda^6}{104}$$

$$t^9 : \frac{2}{9} - \frac{35\lambda^6}{24} - \frac{65\lambda^4}{104}$$

$$t^{11} : \frac{2}{11} + \frac{7\lambda^4}{8} + \frac{\lambda^2}{2}$$

Formules numériques

1) Simple précision

$$\text{Ap.}[\text{Log } x] = \sum_{n=0}^2 K_{2n+1} \left(\frac{x-1}{x+1} \right)^{2n+1}$$

$$10^{-1/4} \leq x \leq 10^{1/4}$$

$$K_1 = 2,000\,000\,004\,354$$

$$K_3 = 0,666\,659\,948\,202$$

$$K_5 = 0,402\,592\,000\,000$$

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 6 \cdot 10^{-11}$$

2) Double précision

$$\text{Ap.}[\text{Log } x] = \sum_{n=0}^5 K_{2n+1} \left(\frac{x-1}{x+1} \right)^{2n+1}$$

$$10^{-1/4} \leq x \leq 10^{1/4}$$

$$K_1 = 1,999\,999\,999\,999\,990\,376$$

$$K_3 = 0,666\,666\,666\,666\,718\,626\,820$$

$$K_5 = 0,399\,999\,999\,919\,862\,351\,304$$

$$K_7 = 0,285\,714\,338\,667\,981\,474\,965$$

$$K_9 = 0,222\,205\,222\,895\,870\,862\,222$$

$$K_{11} = 0,184\,433\,696\,442\,181\,818\,182$$

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 6 \cdot 10^{-20}$$

122 - 111 - 3 - Développements d'Hastings

Formules numériques

Nous retranscrivons les approximations données par Hastings dans « Approximation for Digital Computers ». La première approximation correspond au logarithme décimal. Il suffit de la multiplier

par $\text{Log}_e 10$ pour être ramené à une approximation du Logarithme népérien .

$$\text{Ap. } [\text{Log } x] = \text{Log}_e 10 \sum_{n=0}^4 K_{2n+1} \left(\frac{x-1}{x+1} \right)^{2n+1}$$

$$\text{avec } 10^{-1/2} \leq x \leq 10^{1/2}$$

$$K_1 = 0,868\,591\,718$$

$$K_7 = 0,094\,376\,476$$

$$K_3 = 0,289\,335\,524$$

$$K_9 = 0,191\,337\,714$$

$$K_5 = 0,177\,522\,071$$

$$\text{Ecart } | \text{ fonction - approximation } | \leq 10^{-7}$$

$$\text{Ap. } [\text{Log } (1+x)] = \sum_{n=1}^8 K_n x^n$$

$$\text{avec } 0 \leq x \leq 1$$

$$K_1 = 0,999\,996\,423\,9$$

$$K_5 = 0,167\,654\,071\,1$$

$$K_2 = -0,499\,874\,123\,8$$

$$K_6 = -0,095\,329\,389\,7$$

$$K_3 = 0,331\,799\,025\,8$$

$$K_7 = 0,036\,088\,493\,7$$

$$K_4 = -0,240\,733\,808\,4$$

$$K_8 = -0,006\,453\,544\,2$$

$$\text{Ecart } | \text{ fonction - approximation } | \leq 3 \cdot 10^{-8}$$

122 - 112 Approximation rationnelle .

Nous pouvons mettre $\text{Log } x$ sous la forme $\text{Log } \frac{1+t}{1-t}$ en effectuant le changement de variable : $t = \frac{x-1}{x+1}$

Nous avons vu précédemment que : $|t| \leq 0,072$ si x appartient à l'intervalle $[10^{-1/16}, 10^{1/16}]$

$\text{Log } \frac{1+t}{1-t}$ peut être développé en fraction continue suivant :

$$\text{Log } \frac{1+t}{1-t} = \frac{2t}{1 - \frac{1^2 t^2 / 1.3}{1 - \frac{2^2 t^2 / 3.5}{1 - \dots}}}$$

chaque numérateur successif se met sous la forme :

$$\frac{n^2 t^2}{(2n-1)(2n+1)}$$

La fraction est dite « arrêtée à la $n^{\text{ième}}$ réduite » si $\frac{n^2 t^2}{(2n-1)(2n+1)}$ est négligé par rapport à l'unité.

La $n^{\text{ième}}$ réduite peut être mise sous la forme d'une fraction rationnelle R_n qui est alors une approximation de la fonction.

Choix de la réduite - Erreur de méthode .

Les coefficients des numérateurs successifs sont négatifs: aussi R_n croît avec n . Le logarithme est donc toujours approché par valeur inférieure .

Nous devons avoir :

$$\text{Log } \frac{1+t}{1-t} - R_n(t) \leq \varepsilon$$

Nous choisissons l'ordre de la réduite et vérifions ensuite que cette dernière est suffisante pour obtenir un écart | fonction - approximation | inférieur ou égal à ε .

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

Nous choisissons la troisième réduite :

$$R_3(t) = \frac{2t(15-4t^2)}{15-9t^2}$$

et nous posons :

$$e(t) = \text{Log } \frac{1+t}{1-t} - R_3(t)$$

Nous devons avoir $e(t) \leq \varepsilon$ quand t varie dans tout l'intervalle $[-0,072 ; 0,072]$. Il suffit de chercher une borne supérieure B de $e(t)$ et de vérifier que sa valeur est inférieure à ε .

Pour déterminer B nous écrivons : $e(t) = \int_{-t}^t \left(\frac{d e(t)}{dt} \right) dt$.

Or, la dérivée de $e(t)$ par rapport à t se présente sous la forme d'une fraction rationnelle. Il suffit de majorer cette fraction en remplaçant son dénominateur par un terme plus petit et de le choisir tel que sa valeur soit indépendante de la variable d'intégration afin de le sortir du signe $\int$. Il reste seulement à intégrer le numérateur pour obtenir B .

$$\frac{d e(t)}{dt} = \frac{72 t^6}{(1-t^2)(15-9t^2)^2} = \frac{72}{15^2} \frac{t^6}{(1-t^2) \left(1 - \frac{9}{15} t^2\right)^2}$$

La dérivée est positive . Le maximum de $e(t)$ est obtenu à l'extrémité de l'intervalle soit pour $|t| = 0,072$.

De plus, $e(t)$ est impair et nul pour $t=0$. Il suffit d'étudier $e(t)$ sur l'intervalle $[0, t]$

$$e(t) = \frac{72}{15^2} \int_0^t \frac{u^6 du}{(1-u^2) \left(1 - \frac{9}{15} u^2\right)^2}$$

Si nous remplaçons $(1-u^2)(1-\frac{9}{15}u^2)^2$ par $(1-t_{\max}^2)^3$, le dénominateur est diminué 2 fois : l'une par la substitution de 1 à $\frac{9}{15}$, l'autre, par celle de $t_{\max}^2$ à u^2 . Il en résulte :

$$e(t) < B = \frac{72}{15^2} \frac{1}{(1-t_{\max}^2)^3} \frac{t_{\max}^7}{7}$$

Pour $t_{\max} = 0,072$, nous avons $B < 4,6 \cdot 10^{-10} < \varepsilon$. Ainsi $R_3(t) = \frac{2t(15-4t^2)}{15-9t^2}$ permet d'obtenir

$\text{Log } \frac{1+t}{1-t}$ avec une erreur absolue inférieure à : $4,6 \cdot 10^{-10}$.

Pour réaliser un gain d'opérations, nous écrivons $R_3(t)$ sous la forme :

$$R_3(t) = \text{Ap. } \left[\text{Log } \frac{1+t}{1-t} \right] \frac{t}{A-t^2} \frac{B}{C-t^2}$$

2) Double précision ($\varepsilon = 0,5 \cdot 10^{-18}$)

Nous choisissons la sixième réduite :

$$R_6(t) = \frac{2t(1155 - 1190t^2 + 231t^4)}{1155 - 1575t^2 + 525t^4 - 25t^6}$$

$$e(t) = \text{Log} \frac{1+t}{1-t} - R_6(t)$$

Pour déterminer une borne supérieure B de $e(t)$ nous utilisons le même procédé qu'en simple précision.

$$\frac{d e(t)}{dt} = \frac{12800 t^{12}}{(1-t^2)(1155 - 1575t^2 + 525t^4 - 25t^6)^2}$$

La dérivée est positive. Le maximum de $e(t)$ est atteint quand $t = 0,072$

$$e(t) = \frac{12800}{(1155)^2} \int_0^t \frac{u^{12} du}{(1-u^2)(1 - \frac{315}{231}u^2 + \frac{105}{231}u^4 - \frac{5}{231}u^6)^2}$$

Nous obtenons pour B la valeur :

$$e(t) < B = \frac{12800}{(1155)^2} \frac{1}{(1-t_{\max}^2) \left(1 - \frac{315}{231}t_{\max}^2\right)^2} \frac{t_{\max}^{13}}{13}$$

Pour $t_{\max} = 0,072$, nous avons $B = 10^{-18}$

La sixième réduite n'est donc pas tout à fait suffisante. Toutefois, nous nous en tiendrons à cette approximation afin de limiter le nombre d'opérations.

Par des divisions successives, nous pouvons mettre $R_6(t)$ sous la forme :

$$R_6(t) = \frac{2t}{1-t^2 \left[A + \frac{t^2}{B-t^2 \left(C-t^2 \frac{D}{E+t^2} \right)} \right]}$$

Nous verrons dans la seconde partie que, dans un calcul en double précision, il faut éviter autant que possible les divisions. C'est pourquoi nous donnons la formule numérique qui correspond à la forme initiale.

Formules numériques

1) Simple précision

$$\text{Ap. } \left[\text{Log} \frac{1+t}{1-t} \right] = \frac{t}{A-t^2 \frac{B}{C-t^2}}$$

$$-0,072 \leq t \leq 0,072$$

$$A = 0,500$$

$$B = 0,625$$

$$C = 3,750$$

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 5.10^{-10}$$

2) Double précision

$$Ap\left[\log \frac{1+t}{1-t}\right] = t \frac{\sum_{n=0}^{\infty} K_{2n} t^{2n}}{\sum_{n=0}^{\infty} K_{2n}^* t^{2n}}$$

$$-0,072 \leq t \leq 0,072$$

$$K_0 = 0,962\,500$$

$$K_2 = 0,991\,666\,666 \dots$$

$$K_4 = 0,192\,500$$

$$K_0^* = 0,481\,250$$

$$K_2^* = -0,656\,250$$

$$K_4^* = 0,218\,750$$

$$K_6^* = -0,010\,166\,666 \dots$$

$$\text{Ecart } | \text{ fonction } - \text{ approximation } | \leq 10^{-18}$$

122-12 - Comparaison des différentes approximations

La comparaison porte sur la simple précision mais les conclusions peuvent s'étendre à la double précision.

Sur la précision :

Nous regroupons dans un tableau les résultats obtenus au paragraphe 122 - 11

Nature de l'approximation	Ecart théorique fonction - approximation
Tchebycheff : $\log(1+\lambda X)$	$< 5.10^{-10}$
Tchebycheff : $\log \frac{1+\lambda X}{1-X}$	$< 5.10^{-11}$
Série - Relaxation	$< 6.10^{-11}$
Fraction continue	$< 5.10^{-10}$

Sur le nombre d'opérations

L'intervalle de variation étant le même pour toutes les approximations, nous ne tenons pas compte dans la comparaison du nombre d'opérations nécessaires à la réduction d'intervalle, puisqu'il reste le même quelle que soit l'approximation choisie. Nous envisageons donc seulement le nombre d'opérations à effectuer pour le changement de variable et le calcul de l'approximation.

Nature de l'approximation	additions	multiplications	divisions
Tchebycheff: $\text{Log} (1 + \lambda X)$	8	8	0
Tchebycheff: $\text{Log} \frac{1 + \lambda X}{1 - \lambda X}$	4	5	1
Série - Relaxation	4	4	1
Fraction continue	4	2	3

L'approximation de Tchebycheff de $\text{Log} (1 + \lambda X)$ présente l'avantage de ne requérir aucune division mais la faible convergence de son développement entraîne un trop grand nombre d'opérations par rapport aux autres approximations; elle donne, en outre, un résultat moins précis.

Le développement en série avec relaxation semble plus avantageux que le développement de Tchebycheff de $\text{Log} \frac{1 + \lambda X}{1 - \lambda X}$ puisqu'il nécessite une multiplication de moins tout en donnant la même précision.

Quant à l'approximation rationnelle, elle requiert le même nombre d'opérations que le développement en série mais elle nécessite plus de divisions et donne un résultat moins précis.

122 - 2 Exponentielle

122-21 - Calculs numériques des approximations .

Pour toutes les approximations étudiées, l'argument est préalablement réduit dans l'intervalle $[-\frac{\text{Log}_e 10}{2^q} ; \frac{\text{Log}_e 10}{2^q}]$. Cette forme provient de la méthode employée pour effectuer la réduction. En effet, si l'argument est désigné par A, nous formons :

$$A \log_{10} e = N + f_0$$

N est la partie entière positive, négative ou nulle. f_0 est la partie fractionnaire positive ($0 < f_0 < 1$)

Nous avons alors :

$$A = (N + f_0) \text{Log}_e 10 \quad \text{et} \quad e^A = 10^N \cdot e^{f_0 \text{Log}_e 10}$$

Le calcul sera mené à celui de $e^{f_0 \text{Log}_e 10}$. Un simple décalage permet de tenir compte de la partie entière dans le résultat final.

Pour réduire l'intervalle de variation de f_0 , nous formons :

$$f_{n+1} = f_n - \frac{\varepsilon_n}{2^{n+1}} \quad (\varepsilon_n \text{ est le signe de } f_n)$$

et répétons plusieurs fois l'opération jusqu'à obtenir $n = q - 1$:

$$f_1 = f_0 - \frac{1}{2} \quad |f_1| < \frac{1}{2}$$

$$\begin{aligned} f_1 > 0 & \quad f_2 = f_1 - \frac{1}{4} \\ f_1 < 0 & \quad f_2 = f_1 + \frac{1}{4} \end{aligned} \quad |f_2| \leq \frac{1}{2^2}$$

$$\begin{aligned} f_{q-1} > 0 & \quad f_q = f_{q-1} - \frac{1}{2^q} \\ f_{q-1} < 0 & \quad f_q = f_{q-1} + \frac{1}{2^q} \end{aligned} \quad |f_q| < \frac{1}{2^q}$$

La somme donne :

$$f_q = f_0 - \left[\frac{1}{2} \pm \frac{1}{4} \dots \pm \frac{1}{2^q} \right]$$

Le résultat final s'écrit :

$$e^A = 10^N \cdot 10^{\left[\frac{1}{2} \pm \frac{1}{4} \dots \pm \frac{1}{2^q} \right]} \cdot e^{f_q \text{Log}_e 10}$$

L'intervalle de variation de $f_q \text{Log}_e 10$ est $\left[-\frac{\text{Log}_e 10}{2^q}, \frac{\text{Log}_e 10}{2^q} \right]$

Le nombre de termes de l'approximation est d'autant plus faible que « q » est grand.

Mais la quantité $\left[\frac{1}{2} \pm \frac{1}{4} \dots \pm \frac{1}{2^q} \right]$ prend 2^{q-1} valeurs différentes et la réduction d'intervalle exige « q » additions. Il y a donc un compromis et nous ne pouvons en fait choisir « q » qu'expérimentalement.

REMARQUE • Pour les approximations polynomiales, en raison de la longueur des calculs, nous avons choisi une seule valeur de q alors que pour les approximations rationnelles nous donnons les résultats correspondant à quatre valeurs de q.

122-211 Approximations polynomiales :

122-211-1 - Développement de Tchebycheff.

Si l'argument x est réduit dans l'intervalle de variation $\left[-\frac{\text{Log}_e 10}{4}, \frac{\text{Log}_e 10}{4} \right]$ nous effectuons le changement de variable :

$$x = \lambda X$$

λ est un facteur numérique qui nous permet d'obtenir pour la nouvelle variable X l'intervalle $[-1, 1]$:

$$\lambda = \frac{\text{Log}_e 10}{4} \quad \text{entraîne} \quad |X| \leq 1$$

Le développement de Tchebycheff est :

$$e^{\lambda X} = \frac{a_0}{2} + \sum_{h=1}^{\infty} a_h T_h(X)$$

Calcul des coefficients a_h

Si $f(X) = e^{\lambda X}$, les dérivées successives pour $X = 0$ sont de la forme :

$$f^{(h)}(0) = \lambda^h$$

d'où :

$$a_h = \frac{1}{2^{h-1} h!} \left[\lambda^h + \sum_{k=1}^{\infty} \frac{\lambda^{h+2k}}{2^{2k} k! (h+1)(h+2) \dots (h+k)} \right]$$

Au coefficient 2 près, nous retrouvons le développement de la fonction de Bessel modifiée de première espèce et d'ordre h entier.

$$a_h = 2 I_h(\lambda)$$

Valeurs des 17 premiers coefficients a_h

Les fonctions de Bessel ont été calculées en triple précision et nous donnons les résultats obtenus après multiplication par 2. Les calculs ont été effectués avec

$$\lambda = \frac{\text{Log}_e 10}{4} = 0,575 \ 646 \ 273 \ 248 \ 511 \ 421 \ 004 \ 497 \ 863 \ 671$$

Le nombre entre parenthèses représente la puissance de 10 par laquelle il faut multiplier la valeur donnée

a_0	=	2,169	147	476	660	513	190	741	361	72	(0)
a_1	=	5,998	216	583	832	556	423	280	864	86	(-1)
a_2	=	8,515	358	576	527	065	094	452	407	94	(-2)
a_3	=	8,113	939	789	138	247	927	892	631	54	(-3)
a_4	=	5,814	431	192	392	325	825	130	262	28	(-4)
a_5	=	3,337	856	589	942	586	745	120	715	64	(-5)
a_6	=	1,598	039	880	644	344	283	951	539	36	(-6)
a_7	=	6,561	060	221	771	276	790	573	373	16	(-8)
a_8	=	2,357	820	784	237	216	928	015	313	12	(-9)
a_9	=	7,533	463	940	742	139	226	755	207	66	(-11)
a_{10}	=	2,166	674	498	582	156	015	835	756	48	(-12)
a_{11}	=	5,665	710	224	336	158	532	460	532	66	(-14)
a_{12}	=	1,358	214	469	170	799	909	203	937	56	(-15)
a_{13}	=	3,005	751	996	644	146	444	813	291	84	(-17)
a_{14}	=	6,177	028	140	762	305	759	770	896	18	(-19)
a_{15}	=	1,184	852	218	673	265	086	390	013	47	(-20)
a_{16}	=	2,129	...	...							(-22)

Développement limité à l'ordre $(n-1)$ - Erreur de méthode.

Il suffit de choisir n de façon que l'inégalité suivante soit vérifiée.

$$\frac{a_n}{1 - \frac{a_{n+1}}{a_n}} \leq \varepsilon$$

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

Le tableau précédent montre que :

$$\frac{a_9}{1 - \frac{a_{10}}{a_9}} < 8 \cdot 10^{-11} < \varepsilon$$

Le développement limité à l'ordre 8 fournit une approximation de $e^{\lambda X}$

$$\text{Ap. } [e^{\lambda X}] = \frac{a_0}{2} + \sum_{h=1}^8 a_h T_h(X)$$

L'erreur de méthode correspondante est inférieure à $8 \cdot 10^{-11}$

2) Double précision ($\varepsilon = 0,5 \cdot 10^{-18}$)

$$\frac{a_{15}}{1 - \frac{a_{16}}{a_{15}}} < 1,2 \cdot 10^{-20} < \varepsilon$$

Une approximation de $e^{\lambda X}$ est :

$$\text{Ap. } [e^{\lambda X}] = \frac{a_0}{2} + \sum_{h=1}^{14} a_h T_h(X)$$

L'erreur de méthode correspondante est inférieure à $1,2 \cdot 10^{-20}$

Formules numériques

1) Simple précision

$$\text{Ap. } \left[e^{\frac{\text{Log}_e 10}{4} X} \right] = \sum_{n=0}^8 K_n X^n \quad |X| \leq 1$$

$K_0 = 1$
$K_1 = 0,575\ 646\ 272\ 57$
$K_2 = 0,165\ 684\ 315\ 84$
$K_3 = 0,031\ 791\ 862\ 03$
$K_4 = 0,004\ 575\ 216\ 29$
$K_5 = 0,000\ 526\ 708\ 67$
$K_6 = 0,000\ 050\ 533\ 67$
$K_7 = 0,000\ 004\ 199\ 08$
$K_8 = 0,000\ 000\ 301\ 80$

$$\text{Ecart} | \text{fonction - approximation} | < 8 \cdot 10^{-11}$$

2) Double précision

$$\text{Ap. } \left[e^{\frac{\text{Log}_e 10}{4} X} \right] = \sum_{n=0}^{14} K_n X^n \quad |X| \leq 1$$

$K_0 = 1$
$K_1 = 0,575\ 646\ 273\ 248\ 511\ 421\ 18$
$K_2 = 0,165\ 684\ 315\ 952\ 449\ 937\ 86$
$K_3 = 0,031\ 791\ 853\ 004\ 585\ 558\ 94$
$K_4 = 0,004\ 575\ 215\ 425\ 438\ 542\ 25$
$K_5 = 0,000\ 526\ 741\ 141\ 792\ 631\ 59$
$K_6 = 0,000\ 050\ 536\ 095\ 873\ 263\ 39$
$K_7 = 0,000\ 004\ 155\ 845\ 035\ 940\ 58$
$K_8 = 0,000\ 000\ 299\ 037\ 088\ 398\ 73$
$K_9 = 0,000\ 000\ 019\ 126\ 621\ 445\ 52$
$K_{10} = 0,000\ 000\ 001\ 101\ 016\ 825\ 90$
$K_{11} = 0,000\ 000\ 000\ 057\ 616\ 746\ 99$
$K_{12} = 0,000\ 000\ 000\ 002\ 763\ 912\ 45$
$K_{13} = 0,000\ 000\ 000\ 000\ 123\ 115\ 60$
$K_{14} = 0,000\ 000\ 000\ 000\ 005\ 060\ 22$

$$\text{Ecart} | \text{fonction - approximation} | < 1,2 \cdot 10^{-20}$$

122- 211 - 2 - Développement en série et Relaxation .

Si l'argument x est réduit dans l'intervalle de variation $\left[-\frac{\text{Log}_e 10}{8}, \frac{\text{Log}_e 10}{8}\right]$

(cf. paragraphe 122-21) , nous posons :

$$x = \lambda X$$

λ est un facteur numérique tel que l'intervalle de variation de X soit $(-1, 1)$.

Nous prenons $\lambda = 0,29$; 0,29 est la valeur approchée par excès de $\frac{\text{Log}_e 10}{8}$. Il en résulte :
 $-1 < X < 1$.

Le développement en série de Mac Laurin est :

$$e^{\lambda X} = \sum_{h=0}^{\infty} \lambda^h \frac{X^h}{h!}$$

Développement limité à l'ordre $(n-1)$ - Relaxation - Erreur de méthode .

L'erreur obtenue en limitant le développement à l'ordre $(n-1)$ est :

$$\left| \sum_{h=n}^{\infty} \frac{\lambda^h}{h!} X^h \right|$$

Cette série est majorée par :

$$\frac{\lambda^n}{n!} \sum_{i=0}^{\infty} \lambda^i = \frac{\lambda^n}{n!} \frac{1}{1-\lambda}$$

Pour choisir n , il suffit que l'inégalité suivante soit vérifiée :

$$\frac{(0,29)^n}{n!} \frac{1}{1-0,29} \leq \varepsilon$$

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

L'inégalité est vérifiée quand n est supérieur ou égal à 9. Le développement limité à l'ordre 8 donne l'approximation :

$$Ap. [e^{\lambda X}] = \sum_{h=0}^8 \frac{\lambda^h}{h!} X^h$$

L'erreur correspondante est inférieure à $6 \cdot 10^{-11}$.

La relaxation du terme en X^8 entraîne une erreur de $\frac{\lambda^8}{8!} \frac{1}{2^7} \simeq 0,97 \cdot 10^{-11} < \varepsilon - 6 \cdot 10^{-11}$. Nous pouvons

donc remplacer $\frac{\lambda^8}{8!} X^8$ par : $\frac{\lambda^8}{8!} \frac{1}{2^7} [256 X^6 - 160 X^4 + 32 X^2 - 1]$

La relaxation du terme en X^8 ne modifie pas le terme en X^7 qui peut lui-même être relaxé. L'erreur correspondante est :

$$\frac{\lambda^7}{7!} \frac{1}{2^6} \simeq \varepsilon - 7 \cdot 10^{-11}$$

$\frac{\lambda^7}{7!} X^7$ est remplacé par $\frac{\lambda^7}{7!} \frac{1}{2^6} [112 X^5 - 56 X^3 + 7 X]$

L'erreur de méthode est inférieure au cumul des erreurs soit $5 \cdot 10^{-10}$

Coefficients corrigés de l'approximation ($\lambda = 0,29$)

Nous revenons à l'ancienne variable x et obtenons :

$$\text{coefficient constant : } 1 - \frac{\lambda^8}{128 \cdot 8!}$$

$$\text{« de } x \text{ : } 1 + \frac{\lambda^6}{64 \cdot 6!}$$

$$\text{« de } x^2 \text{ : } \frac{1}{2!} + \frac{\lambda^6}{4 \cdot 8!}$$

$$\text{« de } x^3 \text{ : } \frac{1}{3!} - \frac{\lambda^4}{8 \cdot 6!}$$

$$\text{« de } x^4 \text{ : } \frac{1}{4!} - \frac{\lambda^4}{6 \cdot 4 \cdot 7!}$$

$$\text{« de } x^5 \text{ : } \frac{1}{5!} + \frac{\lambda^2}{4 \cdot 6!}$$

$$\text{« de } x^6 \text{ : } -\frac{1}{6!} + \frac{\lambda^2}{4 \cdot 7!}$$

2) Double précision ($\varepsilon = 0,5 \cdot 10^{-18}$)

L'inégalité est vérifiée quand « n » est supérieur ou égal à 14. Le développement limité à l'ordre 13 donne l'approximation :

$$Ap. [e^{\lambda X}] = \sum_{h=0}^{13} \frac{\lambda^h}{h!} X^h$$

L'erreur correspondante est inférieure à $4,8 \cdot 10^{-19}$

La relaxation du terme en X^{13} entraîne une erreur de :

$$\frac{\lambda^{13}}{13!} \frac{1}{2^{12}} \approx 4.10^{-21} < \varepsilon - 4.8.10^{-19}$$

Nous pouvons donc remplacer $\frac{\lambda^{13}}{13!} X^{13}$ par

$$\frac{\lambda^{13}}{13! 2^{12}} [13312 X^{11} - 16640 X^9 + 9984 X^7 - 2912 X^5 + 364 X^3 - 13 X]$$

La relaxation du terme en $\frac{\lambda^{12}}{12!} X^{12}$ donne une erreur de l'ordre de 3.10^{-19} légèrement supérieur à ($\varepsilon - 4.8.10^{-19}$). L'inégalité n'est pas vérifiée mais comme l'erreur porte sur la dix-neuvième décimale, la relaxation de ce terme peut se justifier.

$$\frac{\lambda^{12}}{12!} X^{12} \text{ est donc remplacé par :}$$

$$\frac{\lambda^{12}}{12! 2^{12}} [6144 X^{10} - 6912 X^8 + 3584 X^6 - 840 X^4 + 72 X^2 - 1]$$

L'erreur de méthode est inférieure au cumul des erreurs soit 8.10^{-19}

Coefficients corrigés de l'approximation ($\lambda = 0,29$)

$$\text{Coefficient constant : } 1 - \frac{\lambda^{12}}{2048.12!}$$

$$\text{« de } x : 1 - \frac{13 \lambda^{12}}{4096.13!}$$

$$\text{« de } x^2 : \frac{1}{2!} + \frac{72 \lambda^{10}}{2048.12!}$$

$$\text{« de } x^3 : \frac{1}{3!} + \frac{364 \lambda^{10}}{4096.13!}$$

$$\text{« de } x^4 : \frac{1}{4!} - \frac{840 \lambda^8}{2048.12!}$$

$$\text{« de } x^5 : \frac{1}{5!} - \frac{2912 \lambda^8}{4096.13!}$$

$$\text{« de } x^6 : \frac{1}{6!} + \frac{3584 \lambda^6}{2048.12!}$$

$$\text{« de } x^7 : \frac{1}{7!} + \frac{9984 \lambda^6}{4096.13!}$$

$$\text{« de } x^8 : \frac{1}{8!} - \frac{6912 \lambda^4}{2048.12!}$$

$$\text{« de } x^9 : \frac{1}{9!} - \frac{16640 \lambda^4}{4096.13!}$$

$$\text{« de } x^{10} : \frac{1}{10!} + \frac{6144 \lambda^2}{2048.12!}$$

$$\text{« de } x^{11} : \frac{1}{11!} + \frac{13312 \lambda^2}{4096.13!}$$

Formules numériques

1) Simple précision

$$\text{Ap. } [e^{f \log_e 10}] = \sum_{n=0}^8 K_n f^n \quad |f| \leq \frac{1}{8}$$

$K_0 =$	0,999 999 999 991
$K_1 =$	2,302 585 121 403
$K_2 =$	2,650 949 073 929
$K_3 =$	2,034 664 046 836
$K_4 =$	1,171 249 168 171
$K_5 =$	0,541 244 747 772
$K_6 =$	0,207 608 276 654

$$\text{Ecart} | \text{fonction} - \text{approximation} | \leq 5 \cdot 10^{-10}$$

2) Double précision

$$\text{Ap. } [e^{f \log_e 10}] = \sum_{n=0}^{11} K_n f^n \quad |f| \leq \frac{1}{8}$$

$K_0 =$	0,999 999 999 999 999 671
$K_1 =$	2,302 585 092 994 045 683 639
$K_2 =$	2,650 949 055 239 200 523 568
$K_3 =$	2,034 678 592 293 476 876 606
$K_4 =$	1,171 255 148 911 133 308 621
$K_5 =$	0,539 382 929 195 233 365 389
$K_6 =$	0,206 995 849 006 431 367 743
$K_7 =$	0,068 089 365 150 808 038 070
$K_8 =$	0,019 597 656 417 526 212 007
$K_9 =$	0,005 013 920 687 538 031 596
$K_{10} =$	0,001 156 673 443 841 055 448
$K_{11} =$	0,000 242 083 812 899 562 223

$$\text{Ecart} | \text{fonction} - \text{approximation} | \leq 8 \cdot 10^{-19}$$

122 -211 -3 Développement d'Hastings .

Nous remplaçons le calcul de e^x par celui de 10^{N+X} en posant :

$$x \log_{10} e = N + X$$

N partie entière positive, négative ou nulle ;
X la partie décimale positive ($0 \leq X < 1$)

Formule numérique

$$\text{Ap. } [e^x] = 10^N \left(\sum_{n=0}^7 K_n X^n \right) \quad \text{avec } 0 \leq X < 1$$

$K_0 =$	1	$K_4 =$	0,072 951 736 66
$K_1 =$	1,151 292 776 03	$K_5 =$	0,017 421 119 88
$K_2 =$	0,662 730 884 29	$K_6 =$	0,002 554 917 96
$K_3 =$	0,254 393 574 84	$K_7 =$	0,000 932 642 67

$$\text{Ecart} | \text{fonction} - \text{approximation} | \leq 5 \cdot 10^{-8}$$

122-212 - Approximation rationnelle

Nous reprenons les résultats publiés par Kogbetliantz dans la première partie de « Mathematical Methods for Digital Computers » pages 19 et 20 ainsi que dans ses conférences des 20 et 21 septembre 1956 (Voir annexe II [3]) :

$$e^x = \frac{P_n(x)}{P_n(-x)} + \frac{R_n(x)}{P_n(-x)}$$

avec

$$P_n(x) = \sum_{h=0}^n a_{n,h} x^h$$

$$a_{n,h} = \frac{(2n-h)! n!}{h! (n-h)! (2n)!}$$

$$R_n(x) = (-1)^n \frac{x^{2n+1}}{(2n)!} \int_0^1 e^{ux} [u(1-u)]^n du$$

Nous prenons pour approximation de e^x

$$Ap. [e^x] = \frac{P_n(x)}{P_n(-x)}$$

Détermination de l'ordre « n » de l'approximation . Erreur .

La valeur absolue du terme complémentaire $\frac{R_n(x)}{P_n(-x)}$ doit être inférieure à ε dans tout l'intervalle de variation de x afin de vérifier :

$$|\text{fonction} - \text{approximation}| < \varepsilon$$

Il suffit de chercher une borne supérieure $B(n)$ de $\left| \frac{R_n(x)}{P_n(-x)} \right|$ et de choisir « n » tel que : $B(n) < \varepsilon$.

Que x soit positif ou négatif , $|R_n(x)|$ est majoré par :

$$|R_n(x)| < \frac{e^{|x|} |x|^{2n+1}}{(2n)!} \int_0^1 [u(1-u)]^n du$$

Le calcul de l'intégrale donne : $\frac{\sqrt{\pi}}{2^{2n+1}} \frac{n!}{(n+\frac{1}{2})!}$

Si nous introduisons la fonction Γ :

$$(n+\frac{1}{2})! = \Gamma(n+\frac{3}{2}) = \underbrace{(n+\frac{1}{2})(n-\frac{1}{2}) \dots (\frac{3}{2})(\frac{1}{2})}_{(n+1) \text{ termes}} \Gamma(\frac{1}{2})$$

Le facteur de $\Gamma(\frac{1}{2})$ peut s'écrire sous la forme :

$$\frac{(2n+1)!}{2^{n+1}} \frac{1}{2^n n!}$$

Le résultat du calcul de l'intégrale devient :

$$\frac{\sqrt{\pi}}{2^{2n+1}} \frac{n!}{(n+\frac{1}{2})!} = \frac{(n!)^2}{(2n+1)!}$$

après simplification par $\Gamma(\frac{1}{2}) = \sqrt{\pi}$ et 2^{2n+1}

donc

$$|R_n(x)| < \frac{(n!)^2}{(2n)!(2n+1)!} |x|^{2n+1} \cdot e^{|x|}$$

Le développement de $P_n(-x)$ est de la forme :

$$P_n(-x) = 1 - \frac{1}{2}x + \frac{1}{2} \frac{n-1}{2n-1} x^2 - \dots$$

La valeur absolue de chaque terme est inférieure à celle du précédent et les signes sont alternés : nous pouvons minorer $P_n(-x)$ par la quantité $1 - \frac{|x|}{2}$

d'où :

$$\left| \frac{R_n(x)}{P_n(-x)} \right| < \frac{(n!)^2}{(2n)! (2n+1)!} \frac{|x|^{2n+1} e^{|x|}}{1 - \frac{|x|}{2}}$$

Si x est réduit dans l'intervalle $\left[-\frac{\text{Log}_e 10}{2^q}, \frac{\text{Log}_e 10}{2^q}\right]$ (voir paragraphe 122.21) nous obtenons

une borne supérieure de $\left| \frac{R_n(x)}{P_n(-x)} \right|$, désignée par $B(n, q)$ pour x maximum ($x = \frac{\text{Log}_e 10}{2^q}$)

$$\left| \frac{R_n(x)}{P_n(-x)} \right| < B(n, q) = \frac{(n!)^2}{(2n)! (2n+1)!} \frac{\left(\frac{\text{Log}_e 10}{2^q}\right)^{2n+1} \cdot 10^{\frac{1}{2}q}}{\left(1 - \frac{\text{Log}_e 10}{2^{q+1}}\right)}$$

Nous pouvons ainsi dresser le tableau des valeurs de $B(n, q)$

$n \backslash q$	1	2	3	4
2				$< 1.10^{-7}$
3			$< 3.10^{-9}$	$< 2.10^{-11}$
4		$< 7.10^{-10}$	$< 9.10^{-13}$	$< 2.10^{-15}$
5	$< 4.10^{-9}$	$< 6.10^{-13}$	$< 2.10^{-18}$	$< 7.10^{-20}$
6	$< 8.10^{-12}$	$< 4.10^{-16}$	$< 3.10^{-20}$	
7	$< 2.10^{-14}$	$< 2.10^{-19}$		
8	$< 2.10^{-17}$			
9	$< 2.10^{-20}$			

Pour vérifier : $B(n, q) < \varepsilon$, il suffit de choisir « n » égal à

$n = 6, 5, 4, 3$ en simple précision ($\varepsilon = 0,5.10^{-9}$)

$n = 9, 7, 6, 5$ en double précision ($\varepsilon = 0,5.10^{-18}$)

q prenant les valeurs 1, 2, 3, 4.

Formules numériques

1) Simple précision

$$\text{Ap. } [e^x] = \frac{\sum_{n=0}^9 K_n x^n}{\sum_{n=0}^9 K_n (-x)^n} \quad \text{avec } |x| \leq \frac{\text{Log}_e 10}{2}$$

$K_0 = 1$
 $K_1 = 0,5$
 $K_2 = 0,113 \ 636 \ 363 \ 636$
 $K_3 = 0,015 \ 151 \ 515 \ 152$
 $K_4 = 0,001 \ 262 \ 626 \ 263$
 $K_5 = 0,000 \ 063 \ 131 \ 313$
 $K_6 = 0,000 \ 001 \ 503 \ 127$

Ecart | fonction - approximation | $< 8.10^{-12}$

$$Ap. [e^x] = \frac{\sum_{n=0}^5 K_n x^n}{\sum_{n=0}^5 K_n (-x)^n} \quad \text{avec } |x| \leq \frac{\text{Log}_e 10}{4}$$

$$\begin{aligned} K_0 &= 1 \\ K_1 &= 0,5 \\ K_2 &= 0,111\ 111\ 111\ 111 \\ K_3 &= 0,013\ 888\ 888\ 889 \\ K_4 &= 0,000\ 992\ 063\ 492 \\ K_5 &= 0,000\ 033\ 068\ 783 \end{aligned}$$

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 6 \cdot 10^{-13}$$

$$Ap. [e^x] = \frac{\sum_{n=0}^4 K_n x^n}{\sum_{n=0}^4 K_n (-x)^n} \quad \text{avec } |x| \leq \frac{\text{Log}_e 10}{8}$$

$$\begin{aligned} K_0 &= 1 \\ K_1 &= 0,5 \\ K_2 &= 0,107\ 142\ 857\ 143 \\ K_3 &= 0,011\ 904\ 761\ 905 \\ K_4 &= 0,000\ 595\ 238\ 095 \end{aligned}$$

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 9 \cdot 10^{-13}$$

$$Ap. [e^x] = \frac{\sum_{n=0}^3 K_n x^n}{\sum_{n=0}^3 K_n (-x)^n} \quad \text{avec } |x| \leq \frac{\text{Log}_e 10}{16}$$

$$\begin{aligned} K_0 &= 1 \\ K_1 &= 0,5 \\ K_2 &= 0,1 \\ K_3 &= 0,008\ 333\ 333\ 333 \end{aligned}$$

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 2 \cdot 10^{-11}$$

2) Double précision

$$Ap. [e^x] = \frac{\sum_{n=0}^9 K_n x^n}{\sum_{n=0}^9 K_n (-x)^n} \quad \text{avec } |x| \leq \frac{\text{Log}_e 10}{2}$$

$$\begin{aligned} K_0 &= 1 \\ K_1 &= 0,5 \\ K_2 &= 0,117\ 647\ 058\ 823\ 529\ 411\ 765 \\ K_3 &= 0,017\ 156\ 862\ 745\ 098\ 039\ 216 \\ K_4 &= 0,001\ 715\ 686\ 274\ 509\ 803\ 922 \\ K_5 &= 0,000\ 122\ 549\ 019\ 607\ 843\ 137 \\ K_6 &= 0,000\ 006\ 284\ 565\ 108\ 094\ 520 \\ K_7 &= 0,000\ 000\ 224\ 448\ 753\ 860\ 519 \\ K_8 &= 0,000\ 000\ 005\ 101\ 108\ 042\ 285 \\ K_9 &= 0,000\ 000\ 000\ 056\ 678\ 978\ 248 \end{aligned}$$

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 2 \cdot 10^{-20}$$

$$\text{Ap. } [e^x] = \frac{\sum_{n=0}^7 K_n x^n}{\sum_{n=0}^7 K_n (-x)^n} \text{ avec } |x| \leq \frac{\text{Log}_e 10}{4}$$

$$\begin{aligned} K_0 &= 1 \\ K_1 &= 0,5 \\ K_2 &= 0,115\ 384\ 615\ 384\ 615\ 384\ 615 \\ K_3 &= 0,016\ 025\ 641\ 025\ 641\ 025\ 641 \\ K_4 &= 0,001\ 456\ 876\ 456\ 876\ 456\ 876 \\ K_5 &= 0,000\ 087\ 412\ 587\ 412\ 587\ 413 \\ K_6 &= 0,000\ 003\ 237\ 503\ 237\ 503\ 237 \\ K_7 &= 0,000\ 000\ 057\ 812\ 557\ 812\ 558 \end{aligned}$$

$$\text{Ecart } | \text{ fonction } - \text{ approximation } | < 2.10^{-19}$$

$$\text{Ap. } [e^x] = \frac{\sum_{n=0}^8 K_n x^n}{\sum_{n=0}^8 K_n (-x)^n} \text{ avec } |x| \leq \frac{\text{Log}_e 10}{8}$$

$$\begin{aligned} K_0 &= 1 \\ K_1 &= 0,5 \\ K_2 &= 0,113\ 636\ 363\ 636\ 363\ 636\ 364 \\ K_3 &= 0,015\ 151\ 515\ 151\ 515\ 151\ 515 \\ K_4 &= 0,001\ 262\ 626\ 262\ 626\ 262\ 626 \\ K_5 &= 0,000\ 063\ 131\ 313\ 131\ 313\ 131 \\ K_6 &= 0,000\ 001\ 503\ 126\ 503\ 126\ 503 \end{aligned}$$

$$\text{Ecart } | \text{ fonction } - \text{ approximation } | < 3.10^{-20}$$

$$\text{Ap. } [e^x] = \frac{\sum_{n=0}^5 K_n x^n}{\sum_{n=0}^5 K_n (-x)^n} \text{ avec } |x| \leq \frac{\text{Log}_e 10}{16}$$

$$\begin{aligned} K_0 &= 1 \\ K_1 &= 0,5 \\ K_2 &= 0,111\ 111\ 111\ 111\ 111\ 111\ 111 \\ K_3 &= 0,013\ 888\ 888\ 888\ 888\ 888\ 889 \\ K_4 &= 0,000\ 992\ 063\ 492\ 063\ 492\ 063 \\ K_5 &= 0,000\ 033\ 068\ 783\ 068\ 783\ 069 \end{aligned}$$

$$\text{Ecart } | \text{ fonction } - \text{ approximation } | < 7.10^{-20}$$

122 - 22 - Comparaison des différentes approximations .

Pour simplifier l'écriture, nous désignons par R_q l'approximation rationnelle dans laquelle

$$|\text{argument}| \leq \frac{\text{Log}_e 10}{2^q}$$

Sur la précision

En simple précision, les écarts obtenus sont reportés dans le tableau suivant :

Nature de l'approximation	Ecart théorique fonction - approximation
Tchebycheff	$< 8.10^{-11}$
Série - Relaxation	$\leq 5.10^{-10}$
R_1	$< 8.10^{-12}$
R_2	$< 6.10^{-13}$
R_3	$< 9.10^{-13}$
R_4	$< 2.10^{-11}$

Avec les approximations polynomiales, nous pouvons choisir l'ordre « n » de façon que l'écart obtenu soit très proche de l'écart désiré. Il n'en est pas de même avec les approximations rationnelles. En effet, nous avons observé que lorsque x est réduit dans le même intervalle, nous devons faire varier l'ordre de 2 unités pour les premières et d'une seule pour les secondes afin de réaliser un gain de 3 décimales sur la précision. Ainsi pour les approximations polynomiales, nous avons la possibilité d'obtenir une précision intermédiaire (en ne faisant varier n que d'une unité), précision qui peut être suffisante. Pour les approximations rationnelles, la précision augmentera obligatoirement de 3 décimales même si une seule suffit.

Sur le nombre d'opérations

Pour toutes les approximations, l'argument est d'abord réduit dans l'intervalle $[0, \text{Log}_e 10]$. Dans la comparaison, nous ne tenons pas compte du nombre d'opérations nécessaires à cette réduction puisqu'il est le même pour toutes les approximations.

L'intervalle $[0, \text{Log}_e 10]$ est réduit en $[-\frac{\text{Log}_e 10}{2^q}, \frac{\text{Log}_e 10}{2^q}]$, « q » étant variable suivant

les approximations ; cette nouvelle réduction nécessite « q » additions (voir paragraphe 122-21)

En les ajoutant aux opérations exigées par le changement de variable et le calcul de l'approximation, nous obtenons le tableau suivant :

Nature de l'approximation	Additions	Multiplications	Divisions
Tchebycheff	10	9	0
Série - Relaxation	9	7	0
R_1	13	12	1
R_2	12	10	1
R_3	11	8	1
R_4	10	6	1

Ce tableau montre que les opérations requises dans les approximations rationnelles sont dans l'ensemble plus nombreuses. En effet, si nous disposons de deux approximations, l'une polynomiale (P), l'autre rationnelle (R), telles que :

$$\begin{aligned} &| \text{fonction} - P_{n-1} | \text{ et } | \text{fonction} - R_{n-1} | > \varepsilon \\ &| \text{fonction} - P_n | \text{ et } | \text{fonction} - R_n | < \varepsilon \end{aligned}$$

Le passage de l'ordre " $n-1$ " à l'ordre " n " demande le calcul d'un terme supplémentaire pour passer de P_{n-1} à P_n soit 1 addition et 1 multiplication, et de 2 (l'un au numérateur, l'autre au dénominateur) pour passer de R_{n-1} à R_n , soit 2 additions et 2 multiplications. Ainsi, dans le cas de l'approximation rationnelle, nous avons 2 opérations de plus pour obtenir souvent un excédent superflu de précision. Cette approximation présente, en outre, l'inconvénient de nécessiter une division.

Nous observons, enfin, que nous réalisons un gain de 1 addition et 2 multiplications chaque fois que nous réduisons l'intervalle de moitié, mais l'encombrement n'en est pas réduit pour autant car le nombre de constantes précalculées augmente avec " q " (voir paragraphe 122 - 21)

122 - 3 Sinus - Cosinus

122 - 31 Calcul numérique des approximations

Avant le calcul proprement dit, il convient de réduire l'argument dans un intervalle donné ; que la fonction cherchée soit sinus ou cosinus, la détermination du résultat exige la connaissance de un ou plusieurs développements suivant le choix de cet intervalle de réduction. Les développements convergent d'autant plus vite que l'intervalle est étroit, mais, par contre, leur nombre est plus important.

En effet, si l'intervalle de réduction est de la forme $\left[-\frac{\pi}{2^q}, \frac{\pi}{2^q}\right]$ (q est entier et positif), il suffit d'augmenter " q " pour réduire le nombre de termes des développements limités mais, comme le résultat peut varier, au signe près, entre 0 et 1, y compris les bornes, il faut connaître alors 2^{q-1} développements limités pour couvrir tout l'intervalle $(0,1)$.

Par exemple, pour $q = 1$, le résultat est parfaitement déterminé, quelle que soit la valeur de l'argument initial, à l'aide d'un seul développement, soit celui de $\sin \frac{\pi}{2}X$, soit celui de $\cos \frac{\pi}{2}X$ ($|X| \leq 1$). Si maintenant nous voulons accroître la convergence et que nous choisissons pour cela $q = 2$, nous devons connaître deux développements, celui de $\sin \frac{\pi}{4}X$ et celui de $\cos \frac{\pi}{4}X$ ($|X| \leq 1$), pour pouvoir, suivant la valeur initiale de l'argument, déterminer dans tous les cas le résultat. Nous nous en tiendrons à ces deux valeurs de q .

122 - 311 Approximations polynomiales

122 - 311-1 - Développement de Tchebycheff de $\sin \lambda X$ et $\cos \lambda X$

$$122 - 311 - 11 : \lambda = \frac{\pi}{4}$$

L'argument est réduit dans l'intervalle $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$ et nous effectuons le changement de variable :

$$x = \lambda X$$

$$\frac{\pi}{4} \leq x \leq \frac{\pi}{4}$$

Le choix de $\lambda = \frac{\pi}{4}$ donne pour la nouvelle variable X l'intervalle de variation $[-1, 1]$

1) $\cos \lambda X$ est une fonction paire ; les coefficients d'ordre impair sont nuls et le développement de Tchebycheff devient alors :

$$\cos \lambda X = \frac{a_0}{2} + \sum_{h=1}^{\infty} a_{2h} T_{2h}(X) \text{ avec } |X| \leq 1$$

Les dérivées successives d'ordre pair pour $X = 0$ sont de la forme :

$$f^{(2h)}(0) = (-1)^h \lambda^{2h}$$

d'où :

$$a_{2h} = \frac{1}{2^{2h-1} (2h)!} \left[(-1)^h \lambda^{2h} + \sum_{k=1}^{\infty} \frac{(-1)^{h+k} \lambda^{2(h+k)}}{2^{2k} k! (2h+1) \dots (2h+k)} \right]$$

Au coefficient 2 près, nous retrouvons le développement des fonctions de Bessel de 1^{ère} espèce d'ordre $2h$:

$$a_{2h} = 2(-1)^h J_{2h}(\lambda)$$

2) $\sin \lambda X$ est une fonction impaire ; les coefficients d'ordre pair sont nuls .

$$\sin \lambda X = \sum_{h=0}^{\infty} a_{2h+1} T_{2h+1}(X) \text{ avec } |X| \leq 1$$

$$f^{(2h+1)}(0) = (-1)^h \lambda^{2h+1}$$

d'où :

$$a_{2h+1} = \frac{1}{2^{2h} (2h+1)!} \left[(-1)^h \lambda^{2h+1} + \sum_{k=1}^{\infty} \frac{(-1)^{h+k} \lambda^{2h+2k+1}}{2^{2k} k! (2h+2) \dots (2h+k+1)} \right]$$

$$a_{2h+1} = 2(-1)^h J_{2h+1}(\lambda)$$

Valeurs des 10 premiers coefficients $a_{2h}(\cos \lambda X)$

Les fonctions de Bessel ont été calculées en triple précision avec $\lambda = \frac{\pi}{4} =$

0,785 398 163 397 448 309 615 660 846

Le nombre entre parenthèses indique la puissance de 10 par laquelle il faut multiplier la valeur donnée;

a_0	=	1, 703	263	827	409	616	025	400	812	03 (0)
a_2	=	- 1, 464	366	443	908	368	633	207	963	61 (- 1)
a_4	=	1, 921	449	311	814	646	796	907	145	45 (- 3)
a_6	=	- 9, 964	968	489	829	300	068	669	106	28 (- 6)
a_8	=	2, 757	659	560	718	739	518	643	839	38 (- 8)
a_{10}	=	- 4, 739	949	808	164	844	037	442	295	12 (- 11)
a_{12}	=	5, 549	548	541	485	182	740	827	264	20 (- 14)
a_{14}	=	- 4, 709	704	906	517	555	956	603	852	00 (- 17)
a_{16}	=	3, 029	897	608	079	373	133	885	742	52 (- 20)
a_{18}	=	- 1, 528	414	934	214	615	335	826	650	59 (- 23)

Valeurs des 10 premiers coefficients a_{2h+1} ($\sin \lambda X$)

a_1	=	7, 263	756	766	937	346	635	911	874	96 (- 1)
a_3	=	- 1, 942	002	905	320	150	630	592	348	48 (- 2)
a_5	=	1, 516	929	228	510	739	948	116	776	77 (- 4)
a_7	=	- 5, 605	804	684	120	011	043	035	489	18 (- 7)
a_9	=	1, 205	324	167	854	356	035	815	895	25 (- 9)
a_{11}	=	- 1, 694	139	308	709	510	635	277	356	08 (- 12)
a_{13}	=	1, 677	809	317	596	605	131	822	242	60 (- 15)
a_{15}	=	- 1, 233	791	086	579	835	763	583	514	92 (- 18)
a_{17}	=	7, 002	577	738	088	537	326	028	238	02 (- 22)
a_{19}	=	- 3, 160	267	932	488	318	862	015	009	00 (- 25)

Développements limités aux ordres $(2n-2)$ et $(2n-1)$ - Erreur de méthode -

Il suffit de choisir n de façon que l'inégalité : $|a_n| < \varepsilon$ soit vérifiée.

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

Les tableaux précédents donnent :

pour cosinus $|a_{10}| < 5 \cdot 10^{-11} < \varepsilon$

pour sinus $|a_{11}| < 2 \cdot 10^{-12} < \varepsilon$

d'où les approximations :

$$\text{Ap. } [\cos \lambda X] = \frac{a_0}{2} + \sum_{h=1}^4 \frac{a_{2h}}{2h} T_{2h}(X)$$

L'erreur de méthode correspondante est inférieure à $5 \cdot 10^{-11}$

$$\text{Ap. } [\sin \lambda X] = \sum_{h=0}^4 \frac{a_{2h+1}}{2h+1} T_{2h+1}(X)$$

L'erreur de méthode correspondante est inférieure à $2 \cdot 10^{-12}$

2) Double précision ($\varepsilon = 0,5 \cdot 10^{-18}$)

pour cosinus $|a_{18}| < 3 \cdot 10^{-20} < \varepsilon$

pour sinus $|a_{17}| < 7 \cdot 10^{-22} < \varepsilon$

d'où les approximations :

$$Ap. [\cos \lambda X] = \frac{a_0}{2} + \sum_{h=1}^7 a_{2h} T_{2h}(X)$$

L'erreur de méthode correspondante est inférieure à 3.10^{-20}

$$Ap. [\sin \lambda X] = \sum_{h=0}^7 a_{2h+1} T_{2h+1}(X)$$

L'erreur de méthode correspondante est inférieure à 7.10^{-22}

Formules numériques

1) Simple précision

$$Ap. [\cos \frac{\pi}{4} X] = \sum_{n=0}^4 K_{2n} X^{2n} \quad |X| \leq 1$$

$K_0 =$	0, 999	999	999	953
$K_2 =$	-0, 308	425	135	160
$K_4 =$	0, 015	854	325	237
$K_6 =$	-0, 000	325	938	600
$K_8 =$	0, 000	003	529	804

$$\text{Ecart} \mid \text{fonction} - \text{approximation} \mid < 5.10^{-11}$$

$$Ap. [\sin \frac{\pi}{4} X] = \sum_{n=0}^4 K_{2n+1} X^{2n+1} \quad |X| \leq 1$$

$K_1 =$	0, 785	398	163	379
$K_3 =$	-0, 080	745	511	815
$K_5 =$	0, 002	490	392	478
$K_7 =$	-0, 000	036	571	417
$K_9 =$	0, 000	000	308	563

$$\text{Ecart} \mid \text{fonction} - \text{approximation} \mid < 2.10^{-12}$$

2) Double précision

$$Ap. [\cos \frac{\pi}{4} X] = \sum_{n=0}^7 K_{2n} X^{2n} \quad |X| \leq 1$$

$K_0 =$	0, 999	999	999	999	999	999	970
$K_2 =$	-0, 308	425	137	534	042	452	958
$K_4 =$	0, 015	854	344	243	815	419	342
$K_6 =$	-0, 000	325	991	886	926	737	786
$K_8 =$	0, 000	003	590	860	446	028	362
$K_{10} =$	-0, 000	000	024	611	364	034	253
$K_{12} =$	0, 000	000	000	115	005	120	719
$K_{14} =$	-0, 000	000	000	000	385	819	025

$$\text{Ecart} \mid \text{fonction} - \text{approximation} \mid < 3.10^{-20}$$

$$Ap. \left[\sin \frac{\pi}{4} X \right] = \sum_{n=0}^7 K_{2n+1} X^{2n+1} \quad |X| \leq 1$$

$K_1 =$	0, 785	398	163	397	448	309	604
$K_3 =$	0, 080	745	512	188	280	781	135
$K_5 =$	0, 002	490	394	570	192	712	152
$K_7 =$	0, 000	036	576	204	182	126	909
$K_9 =$	0, 000	000	313	361	688	869	964
$K_{11} =$	0, 000	000	001	757	247	355	926
$K_{13} =$	0, 000	000	000	006	948	111	082
$K_{15} =$	0, 000	000	000	000	020	214	432

$$\text{Ecart} \mid \text{fonction - approximation} \mid < 7,10^{-22}$$

$$122 - 311 - 12 : \lambda = \frac{\pi}{2}$$

L'argument est réduit dans l'intervalle $\left[-\frac{\pi}{2}, \frac{\pi}{2} \right]$

Nous obtenons les mêmes résultats que précédemment (voir paragraphe 122-311-11) en remplaçant

$$\frac{\pi}{4} \text{ par } \frac{\pi}{2}$$

Valeurs des 11 premiers coefficients $a_{2b} (\cos \lambda X)$

Les fonctions de Bessel ont été calculées avec : $\lambda = \frac{\pi}{2} =$

1,570 796 326 794 896 619 231 321 69

Le nombre entre parenthèses représente la puissance de 10 par laquelle il faut multiplier la valeur donnée.

$a_0 =$	9, 440	024	315	364	695	348	953	367	70	(- 1)
$a_2 =$	4, 994	032	582	704	070	874	009	136	24	(- 1)
$a_4 =$	2, 799	207	961	754	761	751	229	529	52	(- 2)
$a_6 =$	5, 966	951	965	488	464	992	753	518	48	(- 4)
$a_8 =$	6, 704	394	869	916	840	150	086	488	22	(- 6)
$a_{10} =$	4, 653	229	589	731	952	901	092	964	36	(- 8)
$a_{12} =$	2, 193	457	658	956	733	174	654	121	04	(- 10)
$a_{14} =$	7, 481	648	701	033	645	762	263	723	10	(- 13)
$a_{16} =$	1, 932	297	845	863	327	582	068	188	93	(- 15)
$a_{18} =$	3, 910	170	121	632	590	334	884	818	20	(- 18)
$a_{20} =$	6, 367	040	115	833	800	475	768	418	88	(- 21)

Valeurs des 10 premiers coefficients $a_{2b+1} (\sin \lambda X)$

$a_1 =$	1, 133	648	177	811	747	875	422	489	93	(0)
$a_3 =$	- 1, 380	717	765	871	921	035	362	630	84	(- 1)
$a_5 =$	4, 490	714	246	554	917	912	665	723	00	(- 3)
$a_7 =$	- 6, 770	770	584	215	249	116	409	323	50	(- 5)
$a_9 =$	5, 891	295	330	289	313	335	270	070	40	(- 7)
$a_{11} =$	- 3, 338	059	408	918	622	321	199	707	68	(- 9)
$a_{13} =$	1, 329	702	838	448	970	385	255	862	94	(- 11)
$a_{15} =$	3, 927	499	587	179	672	829	122	699	70	(- 14)
$a_{17} =$	8, 945	260	115	941	205	635	594	435	32	(- 17)
$a_{19} =$	- 1, 618	968	996	692	140	662	389	945	94	(- 19)

Développements limités aux ordres $(2n-2)$ et $(2n-1)$ - Erreur de méthode -

Il suffit de choisir "n" de façon que l'inégalité : $|a_n| < \varepsilon$ soit vérifiée

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

Les tableaux précédents donnent :

$$\text{pour cosinus } |a_{12}| < 2,2 \cdot 10^{-10} < \varepsilon$$

$$\text{pour sinus } |a_{13}| < 1,4 \cdot 10^{-11} < \varepsilon$$

d'où les approximations :

$$\text{Ap. } [\cos \lambda X] = \frac{a_0}{2} + \sum_{h=1}^5 a_{2h} T_{2h}(X)$$

L'erreur de méthode est inférieure à $2,2 \cdot 10^{-10}$

$$\text{Ap. } [\sin \lambda X] = \sum_{h=0}^5 a_{2h+1} T_{2h+1}(X)$$

L'erreur de méthode est inférieure à $1,4 \cdot 10^{-11}$

2) Double précision ($\varepsilon = 0,5 \cdot 10^{-18}$)

$$\text{pour cosinus } |a_{20}| < 6,4 \cdot 10^{-21} < \varepsilon$$

$$\text{pour sinus } |a_{19}| < 1,7 \cdot 10^{-19} < \varepsilon$$

d'où les approximations :

$$\text{Ap. } [\cos \lambda X] = \frac{a_0}{2} + \sum_{h=1}^9 a_{2h} T_{2h}(X)$$

L'erreur de méthode est inférieure à $6,4 \cdot 10^{-21}$

$$\text{Ap. } [\sin \lambda X] = \sum_{h=0}^8 a_{2h+1} T_{2h+1}(X)$$

L'erreur de méthode est inférieure à $1,7 \cdot 10^{-19}$

Formules numériques

1) Simple précision

$$\text{Ap. } \left[\cos \frac{\pi}{2} X \right] = \sum_{n=0}^5 K_{2n} X^{2n} \quad |X| \leq 1$$

$$K_0 = 0,999 \ 999 \ 999 \ 780$$

$$K_2 = -1,233 \ 700 \ 534 \ 270$$

$$K_4 = 0,258 \ 669 \ 322 \ 472$$

$$K_6 = -0,020 \ 862 \ 687 \ 548$$

$$K_8 = 0,000 \ 917 \ 723 \ 882$$

$$K_{10} = -0,000 \ 023 \ 824 \ 536$$

$$\text{Ecart } | \text{fonction} - \text{approximation} | < 2,2 \cdot 10^{-10}$$

$$Ap. \left[\sin \frac{\pi}{2} X \right] = \sum_{n=0}^5 K_{2n+1} X^{2n+1} \quad |X| \leq 1$$

$K_1 =$	1,570	796	326	626
$K_3 =$	-0,645	964	092	688
$K_5 =$	0,079	692	587	424
$K_7 =$	-0,004	681	620	352
$K_9 =$	0,000	160	217	088
$K_{11} =$	-0,000	003	418	112

$$\text{Ecart} \mid \text{fonction - approximation} \mid < 1,4 \cdot 10^{-11}$$

2) Double précision

$$Ap. \left[\cos \frac{\pi}{2} X \right] = \sum_{n=0}^9 K_{2n} X^{2n} \quad |X| \leq 1$$

$K_0 =$	0,999	999	999	999	999	999	994
$K_2 =$	-1,233	700	550	136	169	826	079
$K_4 =$	0,253	669	507	901	047	971	531
$K_6 =$	-0,020	863	480	763	352	421	696
$K_8 =$	0,000	919	260	274	835	919	991
$K_{10} =$	-0,000	025	202	042	359	959	684
$K_{12} =$	0,000	000	471	087	448	076	240
$K_{14} =$	-0,000	000	006	386	561	094	481
$K_{16} =$	0,000	000	000	065	623	847	995
$K_{18} =$	-0,000	000	000	000	512	513	818

$$\text{Ecart} \mid \text{fonction - approximation} \mid < 6,4 \cdot 10^{-21}$$

$$Ap. \left[\sin \frac{\pi}{2} X \right] = \sum_{n=0}^8 K_{2n+1} X^{2n+1} \quad |X| \leq 1$$

$K_1 =$	1,570	796	326	794	896	616	150
$K_3 =$	-0,645	964	097	506	246	068	725
$K_5 =$	0,079	692	626	246	163	788	874
$K_7 =$	-0,004	681	754	135	292	622	960
$K_9 =$	0,000	160	441	184	674	323	159
$K_{11} =$	-0,000	003	598	842	947	169	447
$K_{13} =$	0,000	000	056	921	285	353	901
$K_{15} =$	-0,000	000	000	668	396	586	459
$K_{17} =$	0,000	000	000	005	862	365	670

$$\text{Ecart} \mid \text{fonction - approximation} \mid < 1,7 \cdot 10^{-19}$$

122 - 311 - 2 - Développement en série et Relaxation

L'argument est réduit dans l'intervalle $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$. En vue de la relaxation, nous effectuons le changement de variable :

$$x = \lambda X \quad \text{avec} \quad -\frac{\pi}{4} \leq x \leq \frac{\pi}{4}$$

Si nous prenons $\lambda = \frac{\pi}{4}$, l'intervalle de variation de X est $[-1, 1]$

Les développements en série de Mac Laurin sont :

$$\cos \lambda X = \sum_{h=0}^{\infty} (-1)^h \frac{\lambda^{2h}}{(2h)!} X^{2h} \quad \text{avec } |X| \leq 1$$

$$\sin \lambda X = \sum_{h=0}^{\infty} (-1)^h \frac{\lambda^{2h+1}}{(2h+1)!} X^{2h+1} \quad \text{avec } |X| \leq 1$$

Développements limités aux ordres $(2n-2)$ et $(2n-1)$ Troncation . Erreur de méthode

Les séries sont alternées, donc les erreurs :

$$\left| \sum_{h=n}^{\infty} (-1)^h \frac{\lambda^{2h}}{(2h)!} X^{2h} \right| \quad \text{et} \quad \left| \sum_{h=n}^{\infty} (-1)^h \frac{\lambda^{2h+1}}{(2h+1)!} X^{2h+1} \right|$$

sont bornées supérieurement par la valeur maximale du premier terme négligé dans chacun des développements. Il suffit alors de choisir "n" de façon que les inégalités suivantes soient vérifiées :

$$\frac{\lambda^{2n}}{(2n)!} \leq \varepsilon \quad \text{pour cosinus}$$

$$\frac{\lambda^{2n+1}}{(2n+1)!} \leq \varepsilon \quad \text{pour sinus}$$

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

Chaque inégalité est vérifiée quand "n" est supérieur ou égal à 6 . Les approximations correspondantes sont :

$$\text{Ap. } [\cos \lambda X] = \sum_{h=0}^5 (-1)^h \frac{\lambda^{2h}}{(2h)!} X^{2h}$$

L'erreur est inférieure à $1,2 \cdot 10^{-10}$

$$\text{Ap. } [\sin \lambda X] = \sum_{h=0}^5 (-1)^h \frac{\lambda^{2h+1}}{(2h+1)!} X^{2h+1}$$

L'erreur est inférieure à $7 \cdot 10^{-12}$

Chacun de ces développements limités peut être réduit d'un terme par relaxation du dernier.

La relaxation du terme : $-\frac{\lambda^{10}}{10!} X^{10}$ entraîne une erreur de : $\frac{\lambda^{10}}{10!} \frac{1}{2^9} < 5 \cdot 10^{-11} < \varepsilon = 1,2 \cdot 10^{-10}$

Ce terme est alors remplacé par : $-\frac{\lambda^{10}}{2^9 \cdot 10!} [1280 X^8 - 1120 X^6 + 400 X^4 - 50 X^2 + 1]$

La relaxation du terme corrigé en X^9 donne une erreur de $3 \cdot 10^{-8}$. Comme l'erreur est supérieure à ε , ce terme ne peut pas être relaxé.

La relaxation du terme : $-\frac{\lambda^{11}}{11!} X^{11}$ donne une erreur de : $\frac{\lambda^{11}}{11!} \frac{1}{2^{10}} < 2 \cdot 10^{-12} < \varepsilon = 7 \cdot 10^{-12}$

Ce terme est remplacé par : $-\frac{\lambda^{11}}{11! 2^{10}} [2816 X^9 - 2816 X^7 + 1232 X^5 - 220 X^3 + 11 X]$

La relaxation du terme corrigé en X^9 donne une erreur de $1,3 \cdot 10^{-9} > \varepsilon = 7 \cdot 10^{-12}$.

Ce terme ne peut pas être relaxé.

Coefficients corrigés des approximations ($\lambda = \frac{\pi}{4}$)

$Ap_0[\cos \lambda X] = Ap_0[\cos x]$	$Ap_0[\sin \lambda X] = Ap_0[\sin x]$
Coefficient constant : $1 - \frac{\lambda^{10}}{2^9 \cdot 10!}$	Coefficient de x : $1 - \frac{11 \lambda^{10}}{2^{10} \cdot 11!}$
« de x^2 : $-\frac{1}{2!} + \frac{50 \lambda^8}{2^9 \cdot 10!}$	« de x^3 : $-\frac{1}{3!} + \frac{220 \lambda^8}{2^{10} \cdot 11!}$
« de x^4 : $\frac{1}{4!} - \frac{400 \lambda^6}{2^9 \cdot 10!}$	« de x^5 : $\frac{1}{5!} - \frac{1232 \lambda^6}{2^{10} \cdot 11!}$
« de x^6 : $-\frac{1}{6!} + \frac{1120 \lambda^4}{2^9 \cdot 10!}$	« de x^7 : $-\frac{1}{7!} + \frac{2816 \lambda^4}{2^{10} \cdot 11!}$
« de x^8 : $\frac{1}{8!} - \frac{1280 \lambda^2}{2^9 \cdot 10!}$	« de x^9 : $\frac{1}{9!} - \frac{2816 \lambda^2}{2^{10} \cdot 11!}$

L'erreur de méthode est inférieure au cumul des erreurs, soit :

$$2 \cdot 10^{-10} \text{ pour } Ap_0[\cos x]$$

$$9 \cdot 10^{-12} \text{ pour } Ap_0[\sin x]$$

2) Double précision ($\varepsilon = 0,5 \cdot 10^{-18}$)

Les inégalités précédentes sont respectivement vérifiées quand "n" est supérieur ou égal à 10 et 9.

Les approximations correspondantes sont :

$$Ap. [\cos \lambda X] = \sum_{h=0}^9 (-1)^h \frac{\lambda^{2h}}{(2h)!} X^{2h}$$

L'erreur est inférieure à $3,5 \cdot 10^{-21}$

$$Ap. [\sin \lambda X] = \sum_{h=0}^8 (-1)^h \frac{\lambda^{2h+1}}{(2h+1)!} X^{2h+1}$$

L'erreur est inférieure à $9 \cdot 10^{-20}$

$Ap. [\cos \lambda X]$ peut être réduit de deux termes par relaxation des deux derniers :

La relaxation du terme : $-\frac{\lambda^{18}}{18!} X^{18}$ entraîne une erreur de : $\frac{\lambda^{18}}{18!} \frac{1}{2^{17}} < 1,5 \cdot 10^{-23} < \varepsilon - 3,5 \cdot 10^{-21}$

Ce terme est remplacé par : $-\frac{\lambda^{18}}{2^{17} \cdot 18!} \left[589\,824 X^{16} - 1\,105\,920 X^{14} + 1\,118\,208 X^{12} - 658\,944 X^{10} + 228\,096 X^8 - 44\,352 X^6 + 4\,320 X^4 - 162 X^2 + 1 \right]$

Le terme corrigé en X^{16} est alors : $K'_{16} X^{16}$ avec $K'_{16} = \frac{\lambda^{16}}{16!} - \frac{589\,824 \lambda^{18}}{2^{17} \cdot 18!}$

La relaxation de $K'_{16} X^{16}$ donne une erreur de $3 \cdot 10^{-20}$ inférieure à $\varepsilon - 3,5 \cdot 10^{-21}$.

Ce terme est alors remplacé par : $\frac{K'_{16}}{2^{16}} \left[131\,072 X^{14} - 212\,992 X^{12} + 180\,224 X^{10} - 84\,480 X^8 + 21\,504 X^6 - 2688 X^4 + 128 X^2 - 1 \right]$

La relaxation du terme d'ordre X^{14} donne une erreur de $2 \cdot 10^{-16} > \varepsilon - 3,5 \cdot 10^{-21}$:
ce terme ne peut pas être relaxé.

$Ap. [\sin \lambda X]$ ne peut être réduit que d'un seul terme.

La relaxation du terme : $\frac{\lambda^{17}}{17!} X^{17}$ donne une erreur de : $7 \cdot 10^{-22} < \varepsilon - 9 \cdot 10^{-20}$

Ce terme est remplacé par :

$$\frac{\lambda^{17}}{2^{16} \cdot 17!} \left[278\,528 X^{15} - 487\,424 X^{13} + 452\,608 X^{11} - 239\,360 X^9 + 71\,808 X^7 - 11\,424 X^5 + 816 X^3 - 17 X \right]$$

La relaxation du terme corrigé en X^{15} donne une erreur de : $1,2 \cdot 10^{-18} > \varepsilon - 9 \cdot 10^{-20}$

Coefficients corrigés des approximations ($\lambda = \frac{\pi}{4}$)

Ap. [cos x] :

$$\text{coefficient du terme constant : } 1 - \frac{\lambda^{16}}{2^{15} \cdot 16!} + \frac{17 \lambda^{18}}{2^{17} \cdot 18!}$$

$$\text{« de } x^2 : - \frac{1}{2!} + \frac{128 \lambda^{14}}{2^{15} \cdot 16!} - \frac{2142 \lambda^{16}}{2^{17} \cdot 18!}$$

$$\text{« de } x^4 : \frac{1}{4!} - \frac{2688 \lambda^{12}}{2^{15} \cdot 16!} + \frac{44064 \lambda^{14}}{2^{17} \cdot 18!}$$

$$\text{« de } x^6 : - \frac{1}{6!} + \frac{21504 \lambda^{10}}{2^{15} \cdot 16!} - \frac{342\,720 \lambda^{12}}{2^{17} \cdot 18!}$$

$$\text{« de } x^8 : \frac{1}{8!} - \frac{84\,480 \lambda^8}{2^{15} \cdot 16!} + \frac{1\,292\,544 \lambda^{10}}{2^{17} \cdot 18!}$$

$$\text{« de } x^{10} : - \frac{1}{10!} + \frac{180\,224 \lambda^6}{2^{15} \cdot 16!} - \frac{2\,585\,088 \lambda^8}{2^{17} \cdot 18!}$$

$$\text{« de } x^{12} : \frac{1}{12!} - \frac{212\,992 \lambda^4}{2^{15} \cdot 16!} + \frac{2\,715\,648 \lambda^6}{2^{17} \cdot 18!}$$

$$\text{« de } x^{14} : - \frac{1}{14!} + \frac{131\,072 \lambda^2}{2^{15} \cdot 16!} - \frac{1\,253\,376 \lambda^4}{2^{17} \cdot 18!}$$

Ap. [sin x] :

$$\text{coefficient de } x : 1 - \frac{17 \lambda^{16}}{2^{16} \cdot 17!}$$

$$\text{« de } x^3 : - \frac{1}{3!} + \frac{816 \lambda^{14}}{2^{16} \cdot 17!}$$

$$\text{« de } x^5 : \frac{1}{5!} - \frac{11\,424 \lambda^{12}}{2^{16} \cdot 17!}$$

$$\text{« de } x^7 : - \frac{1}{7!} + \frac{71\,808 \lambda^{10}}{2^{16} \cdot 17!}$$

$$\text{« de } x^9 : \frac{1}{9!} - \frac{239\,360 \lambda^8}{2^{16} \cdot 17!}$$

$$\text{« de } x^{11} : - \frac{1}{11!} + \frac{452\,608 \lambda^6}{2^{16} \cdot 17!}$$

$$\text{« de } x^{13} : \frac{1}{13!} - \frac{487\,424 \lambda^4}{2^{16} \cdot 17!}$$

$$\text{« de } x^{15} : - \frac{1}{15!} + \frac{278\,528 \lambda^2}{2^{16} \cdot 17!}$$

L'erreur de méthode est inférieure au cumul des erreurs soit :

$3,5 \cdot 10^{-20}$ pour $Ap. [\cos x]$

$9 \cdot 10^{-20}$ pour $Ap. [\sin x]$

Formules numériques

1) Simple précision

$$Ap. [\cos x] = \sum_{n=0}^4 K_{2n} x^{2n} \quad |x| < \frac{\pi}{4}$$

K_0	= 0,999	999	999	952
K_2	= -0,499	999	996	104
K_4	= 0,041	666	616	135
K_6	= -0,001	388	659	515
K_8	= 0,000	024	376	619

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 2 \cdot 10^{-10}$$

$$Ap. [\sin x] = \sum_{n=0}^4 K_{2n+1} x^{2n+1} \quad |x| \leq \frac{\pi}{4}$$

K_1	= 0,999	999	999	976
K_3	= -0,166	666	665	887
K_5	= 0,008	333	326	259
K_7	= -0,000	198	386	484
K_9	= 0,000	002	713	235

$$\text{Ecart} | \text{fonction} - \text{approximation} | < 9 \cdot 10^{-12}$$

2) Double précision

$$Ap. [\cos x] = \sum_{n=0}^7 K_{2n} x^{2n} \quad \text{avec } |x| \leq \frac{\pi}{4}$$

K_0	= 0,999	999	999	999	999	999	970
K_2	= -0,499	999	999	999	999	993	709
K_4	= 0,041	666	666	666	666	452	459
K_6	= -0,001	388	888	888	886	110	160
K_8	= 0,000	024	801	587	283	884	510
K_{10}	= -0,000	000	275	573	130	986	263
K_{12}	= 0,000	000	002	087	558	248	599
K_{14}	= -0,000	000	000	011	353	385	038

$$\text{Ecart} | \text{fonction} - \text{approximation} | \leq 3,5 \cdot 10^{-20}$$

$$Ap. [\sin x] = \sum_{n=0}^7 K_{2n+1} x^{2n+1} \quad \text{avec } |x| \leq \frac{\pi}{4}$$

$K_1 =$	0,999	999	999	999	999	999	985
$K_3 =$	- 0,166	666	666	666	666	665	477
$K_5 =$	0,008	333	333	333	333	306	342
$K_7 =$	- 0,000	198	412	698	412	423	292
$K_9 =$	0,000	002	755	731	920	911	893
$K_{11} =$	- 0,000	000	025	052	103	828	085
$K_{13} =$	0,000	000	000	160	582	481	943
$K_{15} =$	- 0,000	000	000	000	757	345	818

$$\text{Ecart } | \text{fonction} - \text{approximation} | \leq 9.10^{-20}$$

122 - 311 - 3 Développement d'Hastings

Formule numérique

$$Ap. \left[\sin \frac{\pi}{2} X \right] = \sum_{n=1}^4 K_{2n+1} X^{2n+1} \quad |X| \leq 1$$

$K_1 =$	1,570	796	318	47
$K_3 =$	- 0,645	963	711	06
$K_5 =$	0,079	689	679	28
$K_7 =$	- 0,004	673	765	57
$K_9 =$	0,000	151	484	19

$$\text{Ecart } | \text{fonction} - \text{approximation} | \leq 5.10^{-9}$$

122 - 312 - Approximation rationnelle

Pour calculer sinus et cosinus, nous cherchons une approximation rationnelle de $Tg \frac{x}{2}$

$$Tg \frac{x}{2} = \frac{\sin x}{1 + \cos x} = -i \frac{e^{ix} - e^{-ix}}{2 + e^{ix} + e^{-ix}}$$

Nous reprenons les résultats donnés au paragraphe 122 - 212, en remplaçant x par ix .

$$e^{ix} = \frac{P_n(ix)}{P_n(-ix)} + \frac{R_n(ix)}{P_n(-ix)}$$

$$\text{avec } P_n(ix) = \sum_{h=0}^n a_{n,h} (ix)^h$$

$$a_{n,h} = \frac{(2n-h)! n!}{h! (n-h)! (2n)!}$$

$$R_n(ix) = i \frac{(n!)^2}{(2n)! (2n+1)!} x^{2n+1} e^{i\theta x} \quad (0 < \theta < 1)$$

Dans l'expression de $Tg \frac{x}{2}$, e^{ix} et e^{-ix} sont remplacés par leur développement :

$$\operatorname{Tg} \frac{x}{2} = \frac{P_n(x)}{Q_n(x)} + \frac{(n!)^2}{(2n+1)!(2n)!} \frac{x^{2n+1} \cos \theta x}{(1+\cos x) Q_n(x)}$$

$$\text{avec } P_n(x) = \sum_{h=0}^{\leq \frac{n-1}{2}} (-1)^h a_{n,2h+1} x^{2h+1}$$

$$Q_n(x) = 1 + \sum_{h=1}^{\leq \frac{n}{2}} (-1)^h a_{n,2h} x^{2h}$$

$$\text{Une approximation de } \operatorname{Tg} \frac{x}{2} \text{ est : } t = \operatorname{Ap.} \left[\operatorname{Tg} \frac{x}{2} \right] = \frac{P_n(x)}{Q_n(x)}$$

Si l'argument x est réduit dans l'intervalle $\left[-\frac{\pi}{4}, \frac{\pi}{4} \right]$, les approximations rationnelles de sinus et de cosinus sont :

$$\operatorname{Ap.}[\sin x] = \frac{2t}{1+t^2}$$

$$\operatorname{Ap.}[\cos x] = \frac{1-t^2}{1+t^2}$$

$$\text{avec } |t| \leq \operatorname{Tg} \frac{\pi}{8} < 0,415$$

Détermination de l'ordre "n" de l'approximation - Erreur -

Nous devons choisir "n" tel que :

$$|\sin x - \operatorname{Ap.}[\sin x]| = |\Delta(\sin x)| \leq \varepsilon$$

$$|\cos x - \operatorname{Ap.}[\cos x]| = |\Delta(\cos x)| \leq \varepsilon$$

Différentions les expressions de sinus et cosinus :

$$d(\sin x) = \frac{2(1-t^2)}{(1+t^2)^2} dt < 2 dt$$

$$|d(\cos x)| = \left| -\frac{4t dt}{(1+t^2)^2} \right| < | -4t dt | < 2 dt$$

d'où :

$$|\Delta(\sin x)| < 2 \Delta t \leq \varepsilon$$

$$|\Delta(\cos x)| < 2 \Delta t \leq \varepsilon$$

$$\text{avec } \Delta t = \left| \operatorname{Tg} \frac{x}{2} - \operatorname{Ap.} \left[\operatorname{Tg} \frac{x}{2} \right] \right| = \left| \frac{(n!)^2}{(2n+1)!(2n)!} \frac{x^{2n+1} \cos \theta x}{(1+\cos x) Q_n(x)} \right|$$

Or $Q_n(x)$ reste très voisin de l'unité ; $\cos \theta x$ peut être majoré par 1 et nous obtenons la valeur maximale pour $x = \frac{\pi}{4}$. Il suffit alors de choisir l'ordre n de façon que l'inégalité :

$$\Delta t \leq \frac{(n!)^2}{(2n+1)! (2n)!} \frac{\left(\frac{\pi}{4}\right)^{2n+1}}{1 + \cos \frac{\pi}{4}} \leq \frac{1}{2} \varepsilon \text{ soit vérifiée.}$$

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

L'inégalité précédente est vérifiée quand "n" est supérieur ou égal à 5. L'erreur absolue sur $\operatorname{tg} \frac{x}{2}$ est alors inférieure à $4,2 \cdot 10^{-12}$

t se met sous la forme :

$$t = \frac{a_{5,1} x - a_{5,3} x^3 + a_{5,5} x^5}{1 - a_{5,2} x^2 + a_{5,4} x^4}$$

$$\text{avec : } a_{5,1} = \frac{1}{2} \quad a_{5,2} = \frac{1}{9} \quad a_{5,3} = \frac{1}{72} \quad a_{5,4} = \frac{1}{1008} \quad a_{5,5} = \frac{1}{30240}$$

t est déterminé par la connaissance des cinq variables : x, x^2, x^3, x^4, x^5

Nous cherchons maintenant à réduire ce nombre de variables.

Si nous posons : $x = \pi f$ avec $|f| \leq \frac{1}{4}$ et $t = \frac{f}{T}$, la substitution de la fonction T à t donne :

$$T = \frac{30240 - 3360 \pi^2 f^2 + 30 \pi^4 f^4}{15120 \pi - 420 \pi^3 f^2 + \pi^5 f^4}$$

Le nombre de variables est ainsi réduit à deux au lieu de 5.

Mais par des divisions successives T peut s'écrire sous la forme :

$$T = A + \frac{B}{f^2 + C + \frac{D}{f^2 + E}}$$

La seule variable à connaître est alors f^2

2) Double précision ($\varepsilon = 0,5 \cdot 10^{-18}$)

Cette approximation n'est intéressante que lorsqu'il faut obtenir simultanément sinus et cosinus. Nous verrons ultérieurement qu'en double précision, l'obtention simultanée de sinus et cosinus est difficilement réalisable pour des raisons d'encombrement. Aussi, nous ne poursuivons pas les calculs dans cette précision.

Formules numériques

Simple précision

$$Ap. [\sin \pi f] = \frac{2f}{f^2 + \frac{1}{T}} \quad \text{avec } |f| \leq \frac{1}{4}$$

$$Ap. [\cos \pi f] = \frac{T - \frac{f^2}{T}}{f^2 + \frac{1}{T}}$$

$$T = A + \frac{B}{f^2 + C + \frac{D}{f^2 + E}}$$

$$A = 9,549 \ 296 \ 585 \ 514$$

$$B = 298,004 \ 178 \ 162 \ 763$$

$$C = - 37,912 \ 544 \ 715 \ 624$$

$$D = - 20,781 \ 741 \ 796 \ 623$$

$$E = - 4,642 \ 352 \ 414 \ 158$$

$$\text{Ecart } | \sin \text{ ou } \cos - \text{approximation} | < 8,4 \cdot 10^{-12}$$

122 - 32 Comparaison des approximations

Sur la précision

Quand l'argument est réduit dans l'intervalle $[-\frac{\pi}{4}, \frac{\pi}{4}]$ nous devons connaître, comme nous l'avons vu en tête du paragraphe 122 - 31, les développements de $\sin \lambda X$ et de $\cos \lambda X$. Suivant la valeur initiale de l'argument, nous utilisons l'un ou l'autre. Dans le tableau ci-dessous, nous mentionnons l'écart le plus grand, quitte à en obtenir, dans certains cas, un moindre.

Par exemple, pour l'approximation de Tchebycheff, si le résultat est obtenu par $Ap. [\sin \frac{\pi}{4} X]$, l'écart est inférieur à $2 \cdot 10^{-12}$, s'il l'est par $Ap. [\cos \frac{\pi}{4} X]$, l'écart est inférieur à $5 \cdot 10^{-11}$; c'est ce dernier que nous retiendrons.

Nature de l'approximation	Ecart théorique fonction - approximation
Tchebycheff ($\sin \frac{\pi}{4} X, \cos \frac{\pi}{4} X$)	$< 5 \cdot 10^{-11}$
Tchebycheff ($\sin \frac{\pi}{2} X$)	$< 1,4 \cdot 10^{-11}$
Tchebycheff ($\cos \frac{\pi}{2} X$)	$< 2,2 \cdot 10^{-10}$
Série - Relaxation	$< 1,7 \cdot 10^{-10}$
Rationnelle	$< 8,4 \cdot 10^{-12}$

Sur le nombre d'opérations

Nature de l'approximation		Addition	Multiplication	Division	Résultat
Tchebycheff	$Ap. \left[\sin \frac{\pi}{4} X \right]$	7	7	0	sinus <u>ou</u> cosinus
	$Ap. \left[\cos \frac{\pi}{4} X \right]$	7	6	0	«
Tchebycheff	$Ap. \left[\sin \frac{\pi}{2} X \right]$	7	8	0	«
Tchebycheff	$Ap. \left[\cos \frac{\pi}{2} X \right]$	7	7	0	«
Série-Relaxation	$Ap. \left[\sin x \right]$	7	6	0	«
	$Ap. \left[\cos x \right]$	7	5	0	«
Rationnelle		9	2	4	«
		10	2	5	sinus <u>et</u> cosinus

Dans les approximations polynomiales, si le résultat est compris entre $\frac{\sqrt{2}}{2}$ et 1, nous l'obtenons par $Ap. \left[\cos x \right]$ (voir début paragraphe 122 - 31), en gagnant, dans ce cas, une multiplication .

La réduction de l'argument dans $\left[-\frac{\pi}{4}, \frac{\pi}{4} \right]$ demande une addition supplémentaire par rapport à la réduction dans $\left[-\frac{\pi}{2}, \frac{\pi}{2} \right]$; mais les développements de $\sin \frac{\pi}{4} X$ et de $\cos \frac{\pi}{4} X$ ont alors un terme de moins que ceux de $\sin \frac{\pi}{2} X$ et de $\cos \frac{\pi}{2} X$. Il en résulte un gain de 1 addition et 1 multiplication, soit un gain effectif, pour la réduction dans $\left[-\frac{\pi}{4}, \frac{\pi}{4} \right]$ de 1 multiplication ; le nombre d'additions reste inchangé, quel que soit l'intervalle de réduction .

Dans l'approximation rationnelle, sinus et cosinus ne sont pas obtenus directement puisque nous calculons préalablement la fonction $\frac{1}{\pi} \frac{x}{\operatorname{tg} \frac{x}{2}}$. Ils nécessitent donc un plus grand nombre d'opérations, parmi lesquelles des divisions : cette approximation est donc a priori désavantageuse. Mais avec seulement 1 addition et 1 division supplémentaires, nous pouvons obtenir simultanément sinus et cosinus, ce qui lui donne alors l'avantage sur les autres .

122 - 4 Arc tangente

Il s'agit de la détermination principale, c'est-à-dire, que l'arc qui a pour tangente l'argument donné, est celui compris dans l'intervalle $\left[-\frac{\pi}{2}, \frac{\pi}{2} \right]$

122 - 41 Calcul numérique des approximations

122 - 411 Approximations polynomiales

122 - 411 - 1 Développement de Tchebycheff

Nous prenons pour variable $x = \lambda X$; λ est un facteur numérique à déterminer de façon que l'intervalle de variation de X soit $[-1, 1]$.

Arc $\text{tg}(\lambda X)$ est une fonction impaire ; les coefficients d'ordre pair sont nuls et le développement de Tchebycheff devient :

$$\text{Arc tg}(\lambda X) = \sum_{h=0}^{\infty} a_{2h+1} T_{2h+1}(X)$$

Calcul des coefficients a_{2h+1}

Les dérivées successives d'ordre impair pour $X = 0$ sont de la forme :

$$f^{(2h+1)}(0) = (-1)^h (2h)! \lambda^{2h+1}$$

d'où :

$$a_{2h+1} = \frac{(-1)^h}{2^{2h} (2h+1)!} \left[(2h)! \lambda^{2h+1} + \sum_{k=1}^{\infty} \frac{(2h+2k)! \lambda^{2h+2k+1}}{2^{2k} k! (2h+2) \dots (2h+k+1)} \right]$$

qui peut encore s'écrire :

$$a_{2h+1} = (-1)^h 2 \sum_{k=0}^{\infty} \frac{(2h+2k)!}{k! (2h+k+1)!} \left(\frac{\lambda}{2} \right)^{2h+2k+1}$$

Nous posons $\lambda = \text{tg } 2\alpha$ et procédons par une méthode similaire à celle du paragraphe 122 - 111 - 1 : la limite de la série, quand k tend vers l'infini, devient :

$$\frac{(\text{tg } \alpha)^{2h+1}}{2h+1} \quad \text{soit} \quad a_{2h+1} = (-1)^h 2 \frac{(\text{tg } \alpha)^{2h+1}}{2h+1}$$

x peut varier de $-\infty$ à $+\infty$. Comme nous avons $|X| \leq 1$, $\lambda = \text{tg } 2\alpha$ varie de zéro à l'infini et " α " a pour intervalle de variation $\left(0, \frac{\pi}{4}\right)$. Nous avons vu que, si le développement est limité à l'ordre $(2n-1)$, il suffit de choisir " n " de façon que :

$$|a_{2n+1}| \leq \frac{2}{2n+1} < \varepsilon \quad (\text{correspondant à } \alpha = \frac{\pi}{4})$$

L'erreur de méthode $\frac{2}{2n+1}$ montre que le développement converge très lentement : c'est pourquoi il convient de réduire l'intervalle de variation de x pour avoir $\operatorname{tg}(\alpha_{\max})$ aussi petit que possible.

$$\text{L'erreur de méthode devient alors : } \frac{2[\operatorname{tg}(\alpha_{\max})]^{2n+1}}{2n+1}$$

Réduction de l'intervalle de variation de x .

L'intervalle $(0, \infty)$ est subdivisé en 5 sous-intervalles :

$$I_0 \left(0, \operatorname{tg} \frac{\pi}{20} \right)$$

$$I_1 \left(\operatorname{tg} \frac{\pi}{20}, \operatorname{tg} \frac{3\pi}{20} \right)$$

$$I_2 \left(\operatorname{tg} \frac{3\pi}{20}, \operatorname{tg} \frac{\pi}{4} \right)$$

$$I_3 \left(\operatorname{tg} \frac{\pi}{4}, \operatorname{tg} \frac{3\pi}{8} \right)$$

$$I_4 \left(\operatorname{tg} \frac{3\pi}{8}, \operatorname{tg} \frac{\pi}{2} \right)$$

une fois l'argument x localisé dans I_q ($q = 0, 1, 2, 3, 4$) nous avons, si a et b sont les bornes de l'intervalle I_q :

$$\operatorname{Arc} \operatorname{tg} a \leq \operatorname{Arc} \operatorname{tg} x \leq \operatorname{Arc} \operatorname{tg} b$$

Nous centrons l'intervalle de variation de la fonction sur zéro en retranchant à $\operatorname{Arc} \operatorname{tg} x$ la demi-somme des bornes $\operatorname{Arc} \operatorname{tg} a$ et $\operatorname{Arc} \operatorname{tg} b$:

$$-\frac{1}{2}(\operatorname{Arc} \operatorname{tg} b - \operatorname{Arc} \operatorname{tg} a) \leq \operatorname{Arc} \operatorname{tg} x - \frac{1}{2}(\operatorname{Arc} \operatorname{tg} a + \operatorname{Arc} \operatorname{tg} b) \leq \frac{1}{2}(\operatorname{Arc} \operatorname{tg} b - \operatorname{Arc} \operatorname{tg} a)$$

Nous vérifions, quel que soit l'intervalle I_q , que $\frac{1}{2}(\operatorname{Arc} \operatorname{tg} b - \operatorname{Arc} \operatorname{tg} a)$ reste toujours inférieur ou égal à $\frac{\pi}{16}$. Il suffit alors de poser :

$$\operatorname{Arc} \operatorname{tg} t = \operatorname{Arc} \operatorname{tg} x - \frac{1}{2}(\operatorname{Arc} \operatorname{tg} a + \operatorname{Arc} \operatorname{tg} b)$$

pour obtenir une nouvelle variable " t " dont l'intervalle de variation est réduit à $\left[-\operatorname{tg} \frac{\pi}{16}, \operatorname{tg} \frac{\pi}{16} \right]$

Pour simplifier l'écriture, nous posons :

$$\theta_q = \frac{1}{2}(\operatorname{Arc} \operatorname{tg} a + \operatorname{Arc} \operatorname{tg} b)$$

La valeur de θ_q varie avec q .

REMARQUE 1: Pour $q = 0$, il suffit de prendre $x = t$ puisque $x \leq \operatorname{tg} \frac{\pi}{20}$. La valeur de θ_0 correspondante est nulle.

REMARQUE 2 : Le passage de x à t s'obtient de la façon suivante :

de $\operatorname{tg}(\operatorname{Arc} \operatorname{tg} t) = \operatorname{tg}(\operatorname{Arc} \operatorname{tg} x - \theta_q)$, nous tirons :

$$t = \frac{x - \operatorname{tg} \theta_q}{1 + x \operatorname{tg} \theta_q} = \cotg \theta_q - \frac{1 + \cotg^2 \theta_q}{x + \cotg \theta_q} = a_q - \frac{b_q}{x + a_q}$$

$$\text{avec } a_q = \cotg \theta_q \quad \text{et} \quad b_q = 1 + \cotg^2 \theta_q$$

Développement de Tchebycheff de Arc tg t -

Nous posons : $t = (\operatorname{tg} 2\alpha) \cdot X$ avec $|X| \leq 1$

Nous choisissons pour $\operatorname{tg} 2\alpha$ une valeur telle que l'intervalle de t soit :

$$\left[-\operatorname{tg} \frac{\pi}{16}, \operatorname{tg} \frac{\pi}{16} \right]$$

$$\operatorname{Tg} 2\alpha = \frac{2 \operatorname{tg} \alpha}{1 + \operatorname{tg}^2 \alpha} \geq \operatorname{tg} \frac{\pi}{16}$$

L'inégalité est vérifiée quand $\operatorname{tg} \alpha$ est supérieure ou égale à 0,098. Pour simplifier les calculs suivants, nous prenons 0,1, valeur approchée par excès. La valeur de $\operatorname{tg} 2\alpha$ correspondante est $\frac{20}{99}$.

Le développement de Tchebycheff de Arc tg t devient :

$$\operatorname{Arc} \operatorname{tg} t = \sum_{h=0}^{\infty} a_{2h+1} T_{2h+1} \left(\frac{99t}{20} \right) \quad \left| \frac{99t}{20} \right| < 1 \quad \text{avec} \quad a_{2h+1} = (-1)^h \frac{2}{(2h+1) \cdot 10^{2h+1}}$$

Développement limité à l'ordre $(2n-1)$ - Erreur de méthode -

Il suffit de choisir "n" de façon que :

$$|a_{2n+1}| = \frac{2}{(2n+1) \cdot 10^{2n+1}} < \varepsilon$$

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

L'inégalité précédente est vérifiée quand n est supérieur ou égal à 4. Nous avons l'approximation :

$$\operatorname{Ap.} [\operatorname{Arc} \operatorname{tg} t] = 2 \sum_{h=0}^3 (-1)^h \frac{T_{2h+1} \left(\frac{99t}{20} \right)}{(2h+1) \cdot 10^{2h+1}}$$

L'erreur de méthode correspondante est inférieure à $2,3 \cdot 10^{-10}$

2) Double précision ($\varepsilon = 0,5 \cdot 10^{-18}$)

L'inégalité est vérifiée quand "n" est supérieur ou égal à 9. L'approximation est un développement limité à l'ordre 17. Nous verrons plus loin que le développement en série avec relaxation donne un développement limité à l'ordre 11 avec lequel nous obtenons la même précision. Aussi nous ne poursuivons pas cette étude en double précision.

CONSTANTES PRECALCULEES :

Pour passer des variables x à t et revenir de $\text{Arc tg } t$ à $\text{Arc tg } x$, nous avons 19 constantes précalculées.

1) Valeurs des Arcs Tangentes aux bornes des l_q

$$\frac{\pi}{20} = 0,157 \ 079 \ 632 \ 679$$

$$\frac{3\pi}{20} = 0,471 \ 238 \ 898 \ 038$$

$$\frac{\pi}{4} = 0,785 \ 398 \ 163 \ 397$$

$$\frac{3\pi}{8} = 1,178 \ 097 \ 245 \ 096$$

2) Valeurs des bornes des l_q

$$\text{tg } \frac{\pi}{20} = 0,158 \ 384 \ 440 \ 324$$

$$\text{tg } \frac{3\pi}{20} = 0,509 \ 525 \ 449 \ 494$$

$$\text{tg } \frac{3\pi}{8} = 2,414 \ 213 \ 562 \ 373$$

3) Valeurs des θ_q

$$\theta_1 = \frac{\pi}{10} = 0,314 \ 159 \ 265 \ 359$$

$$\theta_2 = \frac{\pi}{5} = 0,628 \ 318 \ 530 \ 718$$

$$\theta_3 = \frac{5\pi}{16} = 0,981 \ 747 \ 704 \ 247$$

$$\theta_4 = \frac{7\pi}{16} = 1,374 \ 446 \ 785 \ 945$$

4) Valeurs des a_q

$$a_1 = 3,077 \ 683 \ 537 \ 175$$

$$a_2 = 1,376 \ 381 \ 920 \ 471$$

$$a_3 = 0,668 \ 178 \ 637 \ 919$$

$$a_4 = 0,198 \ 912 \ 367 \ 380$$

5) Valeurs des b_q

$b_1 =$	10,472	135	954	999
$b_2 =$	2,894	427	191	000
$b_3 =$	1,446	462	692	172
$b_4 =$	1,039	566	129	897

REMARQUE : Nous pouvons réduire à 11 le nombre de constantes précalculées, en appliquant la formule : $\text{Arc tg } x = \frac{\pi}{2} - \text{Arc tg } \frac{1}{x}$ pour $x > \text{tg } \frac{\pi}{4}$

Formule numérique

$$\text{Ap. [Arc tg t]} = \sum_{n=0}^3 K_{2n+1} t^{2n+1}$$

$$\text{avec } |t| \leq \frac{20}{99} < 0,2$$

$$K_1 = 0,999 \quad 999 \quad 990 \quad 0$$

$$K_3 = -0,333 \quad 330 \quad 049 \quad 8$$

$$K_5 = 0,199 \quad 707 \quad 910 \quad 5$$

$$K_7 = -0,133 \quad 152 \quad 192 \quad 6$$

$$\text{Ecart } | \text{fonction} - \text{approximation} | < 2,3 \cdot 10^{-10}$$

122 - 411 - 2 - Développement en série et Relaxation

L'intervalle de variation de l'argument est $(0,1)$.

Pour x supérieur à 1, nous appliquons la formule :

$$\text{Arc tg } x = \frac{\pi}{2} - \text{Arc tg } \frac{1}{x}$$

Si $x = 0, a b c d \dots$, nous prenons une valeur approchée x_0 de la forme : $x_0 = 0, a5$ et écrivons :

$$\text{Arc tg } x = \text{Arc tg } x_0 + \text{Arc tg } \frac{x - x_0}{1 + x \cdot x_0}$$

$\text{Arc tg } x_0$ est donné par une table précalculée.

L'intervalle de variation de $t = \frac{x - x_0}{1 + x \cdot x_0}$ est alors $(-0,05 ; 0,05)$

$$\text{Arc tg } t \text{ se développe suivant : } \text{Arc tg } t = \sum_{h=0}^{\infty} (-1)^h \frac{t^{2h+1}}{2h+1}$$

Développement limité à l'ordre $(2n-1)$ - Relaxation - Erreur de méthode -

La série est alternée. L'erreur est bornée supérieurement par la valeur absolue maximale du premier terme négligé. Il suffit de choisir "n" de façon que l'inégalité suivante soit vérifiée.

$$\frac{(0,05)^{2n+1}}{2n+1} \leq \varepsilon$$

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

L'inégalité est vérifiée quand "n" est supérieur ou égal à 3. Le développement limité à l'ordre 5 donne l'approximation :

$$Ap_s [\text{Arc tg } t] = \sum_{h=0}^2 (-1)^h \frac{t^{2h+1}}{2h+1}$$

L'erreur correspondante est inférieure à $1,12 \cdot 10^{-10}$

La relaxation du dernier terme entraîne une erreur de : $\frac{(5 \cdot 10^{-2})^5}{5} \frac{1}{2^4} \simeq 4 \cdot 10^{-9} > \varepsilon = 1,12 \cdot 10^{-10}$

Nous ne pouvons donc pas relaxer ce terme.

2) Double précision ($\varepsilon = 0,5 \cdot 10^{-18}$)

Nous remplaçons ε par $\varepsilon' = 0,5 \cdot 10^{-19}$ afin de gagner un chiffre significatif dans le cas où $x_0 = 0,05$. L'inégalité précédente est alors vérifiée quand "n" est supérieur ou égal à 7.

$$Ap_s [\text{Arc tg } t] = \sum_{h=0}^6 (-1)^h \frac{t^{2h+1}}{2h+1}$$

L'erreur est inférieure à $2,05 \cdot 10^{-21}$

Posons $t = 5 \cdot 10^{-2} X$ avec $|X| < 1$. La relaxation du terme en t^{13} entraîne une erreur de :

$$\frac{(5 \cdot 10^{-2})^{13}}{13} \frac{1}{2^{12}} < 2,3 \cdot 10^{-22} < \varepsilon' = 2 \cdot 10^{-21}$$

$$\frac{(5 \cdot 10^{-2})^{13}}{13} X^{13} \text{ peut être remplacé par : } \frac{(5 \cdot 10^{-2})^{13}}{13 \cdot 2^{12}} [13312 X^{11} - 16640 X^9 + 9984 X^7 - 2912 X^5 + 364 X^3 - 13X]$$

La relaxation du nouveau terme en t^{11} donne une erreur de $4,3 \cdot 10^{-19}$ supérieure à ε' . Il ne peut pas être relaxé.

L'erreur de méthode est inférieure au cumul des erreurs soit $2,3 \cdot 10^{-21}$.

Coefficients corrigés de l'approximation ($\lambda = 0,05$)

$$\text{coefficient de } t : 1 - \frac{\lambda^{12}}{2^{12}}$$

$$\text{coefficient de } t^7 : -\frac{1}{7} + \frac{9984 \lambda^6}{13 \cdot 2^{12}}$$

$$\text{« de } t^3 : -\frac{1}{3} + \frac{364 \lambda^{10}}{13 \cdot 2^{12}}$$

$$\text{« de } t^9 : \frac{1}{9} - \frac{16640 \lambda^4}{13 \cdot 2^{12}}$$

$$\text{« de } t^5 : \frac{1}{5} - \frac{2912 \lambda^8}{13 \cdot 2^{12}}$$

$$\text{« de } t^{11} : -\frac{1}{11} + \frac{13312 \lambda^2}{13 \cdot 2^{12}}$$

TABLE DES ARC tg x₀

Pour dresser cette table, nous avons programmé le développement en série d'Arc tg x, le seul dont nous disposons pour le moment. Le calcul a été effectué en triple précision.

Arc tg 0,05 =	0,049	958	395	721	942	761	410	006	287
« 0,15 =	0,148	889	947	609	497	250	586	530	393
« 0,25 =	0,244	978	663	126	864	154	172	082	481
« 0,35 =	0,336	674	819	386	727	181	396	698	632
« 0,45 =	0,422	853	926	132	940	712	966	482	790
« 0,55 =	0,502	843	210	927	860	827	330	882	028
« 0,65 =	0,576	375	220	591	183	680	227	570	479
« 0,75 =	0,643	501	108	793	284	386	802	809	228
« 0,85 =	0,704	494	064	242	217	716	657	480	340
« 0,95 =	0,759	762	754	875	770	828	922	961	190

REMARQUE : Ce programme servira pour vérifier les résultats obtenus en double précision.

Formules numériques

1) Simple précision

$$Ap. [\text{Arc tg } t] = \sum_{n=0}^2 K_{2n+1} t^{2n+1}$$

avec $|t| \leq 0,05$

$$\begin{aligned} K_1 &= 1 \\ K_3 &= -0,333 \quad 333 \quad 333 \quad 333 \\ K_5 &= 0,2 \end{aligned}$$

$$\text{Ecart } | \text{ fonction - approximation } | < 1,2 \cdot 10^{-10}$$

2) Double précision

$$Ap. [\text{Arc tg } t] = \sum_{n=0}^5 K_{2n+1} t^{2n+1}$$

avec $|t| \leq 0,05$

$$\begin{aligned} K_1 &= 0,999 \quad 999 \quad 999 \quad 999 \quad 999 \quad 999 \quad 995 \\ K_3 &= -0,333 \quad 333 \quad 333 \quad 333 \quad 332 \quad 665 \quad 761 \\ K_5 &= 0,199 \quad 999 \quad 999 \quad 997 \quad 863 \quad 769 \quad 531 \\ K_7 &= -0,142 \quad 857 \quad 139 \quad 927 \quad 455 \quad 357 \quad 143 \\ K_9 &= 0,111 \quad 109 \quad 157 \quad 986 \quad 111 \quad 111 \quad 111 \\ K_{11} &= -0,090 \quad 284 \quad 090 \quad 909 \quad 090 \quad 909 \quad 091 \end{aligned}$$

$$\text{Ecart } | \text{ fonction - approximation } | < 2,3 \cdot 10^{-21}$$

122 - 411 - 3 - Développement d'Hastings

Formule numérique

$$Ap, [\text{Arc tg } X] = \sum_{n=0}^7 K_{2n+1} X^{2n+1}$$

avec $|X| \leq 1$

K_1	=	0,999	999	332	9
K_3	= -	0,333	298	560	5
K_5	=	0,199	465	359	9
K_7	= -	0,139	085	335	1
K_9	=	0,096	420	044	1
K_{11}	= -	0,055	909	886	1
K_{13}	=	0,021	861	228	8
K_{15}	= -	0,004	054	058	0

Ecart | fonction - approximation | $\leq 4.10^{-8}$

122 - 412 - Approximation rationnelle

Nous avons vu dans le paragraphe 122 - 411 - 1 que le calcul de Arc tg x peut se ramener à celui de Arc tg t avec $|t| \leq \text{Tg} \frac{\pi}{16}$

Arc tg t peut être développé en fraction continue suivant :

$$\text{Arc tg } t = \frac{t}{1 + \frac{1^2 t^2 / 1.3}{1 + \frac{2^2 t^2 / 3.5}{1 + \dots}}}$$

chaque numérateur successif se met sous la forme : $\frac{n^2 t^2}{(2n-1)(2n+1)}$

La fraction est dite « arrêtée à la n^{ième} réduite » si $\frac{n^2 t^2}{(2n-1)(2n+1)}$ est négligé par rapport à l'unité. La réduite $R_n(t)$ ainsi obtenue est une approximation de Arc tg t.

Choix de la réduite. Erreur.

Les coefficients des numérateurs successifs sont positifs : aussi $R_n(t)$ décroît avec « n ». Arc tg t est donc toujours approché par valeur supérieure.

Nous devons avoir : $R_n(t) - \text{Arc tg } t \leq \varepsilon$

1) Simple précision : ($\varepsilon = 0,5 \cdot 10^{-10}$)

Nous choisissons la cinquième réduite et vérifions qu'elle conduit à une erreur de méthode inférieure à ε

$$R_5(t) = \frac{t(945 + 735t^2 + 64t^4)}{945 + 1050t^2 + 225t^4}$$

$$\text{Posons : } e(t) = R_5(t) - \text{Arc tg } t$$

Il suffit de chercher une borne supérieure de $e(t)$ quand t varie dans tout l'intervalle $\left[-\text{tg } \frac{\pi}{16}, \text{tg } \frac{\pi}{16}\right]$ et de vérifier qu'elle est inférieure à ε . Nous appliquons pour cela la même méthode qu'au paragraphe 122-112

$$\frac{de(t)}{dt} = \frac{64}{63^2} \frac{t^{10}}{(1+t^2)\left(1 + \frac{10}{9}t^2 + \frac{5}{21}t^4\right)^2}$$

La dérivée est positive. Nous obtenons le maximum de $e(t)$ quand $t = \text{tg } \frac{\pi}{16}$. De plus $e(t)$ est impair et nul pour $t = 0$.

$$e(t) = \frac{64}{63^2} \int_0^t \frac{t^{10} dt}{(1+t^2)\left(1 + \frac{10}{9}t^2 + \frac{5}{21}t^4\right)^2}$$

La quantité sous le signe $\int$ est majorée par $\frac{t^{10}}{(1+t^2)^3}$, elle-même majorée par $t^{10}(1-3t^2+6t^4)$ en considérant les 3 premiers termes du développement de $(1+t^2)^{-3}$

Une borne supérieure de $e(t)$ est :

$$\frac{64}{63^2} \frac{\left(\text{tg } \frac{\pi}{16}\right)^{11}}{11} \left[1 - \frac{33}{13} \left(\text{tg } \frac{\pi}{16}\right)^2 + \frac{22}{5} \left(\text{tg } \frac{\pi}{16}\right)^4 \right] < 2,5 \cdot 10^{-11}$$

REMARQUE : Le même calcul d'erreur repris avec :

$$R_4(t) = \frac{t(105 + 55t^2)}{105 + 90t^2 + 9t^4}$$

nous donne $2,5 \cdot 10^{-8}$ comme borne supérieure de $e(t)$. $R_4(t)$ serait donc insuffisant pour obtenir l'écart fixé.

Pour réaliser un gain d'opérations, $R_5(t)$ peut se mettre sous la forme :

$$R_5(t) = t \left[A + \frac{B}{t^2 + C + \frac{D}{t^2 + E}} \right]$$

2) Double précision : ($\varepsilon = 0,5 \cdot 10^{-18}$)

La neuvième réduite est nécessaire pour obtenir la précision désirée. Elle se présente sous la forme :

$$R_9(t) = \frac{\sum_{n=0}^4 K_{2n+1} t^{2n+1}}{\sum_{n=0}^4 K_{2n} t^{2n}}$$

Pour obtenir $R_9(t)$, il faut calculer 10 termes (avec une division en plus) alors que le développement en série avec relaxation demande le calcul de 6 termes seulement. Aussi nous ne poursuivons pas l'étude de l'approximation rationnelle dans cette précision.

Formule numérique

$$Ap. [\text{Arc tg } t] = t \left(A + \frac{B}{t^2 + C + \frac{D}{t^2 + E}} \right)$$

$$\text{avec } |t| \leq \text{tg } \frac{\pi}{16} < 0,2$$

$$A = 0,284 \ 444 \ 444 \ 444$$

$$B = 1,939 \ 259 \ 259 \ 259$$

$$C = 3,116 \ 934 \ 046 \ 346$$

$$D = -0,630 \ 414 \ 367 \ 011$$

$$E = 1,549 \ 732 \ 620 \ 321$$

$$\text{Ecart } | \text{fonction} - \text{approximation} | < 2,5 \cdot 10^{-11}$$

122 - 42 Comparaison des différentes approximations

Sur la précision

Nous obtenons $\text{Arc tg } x$ par : constante précalculée + $Ap. [\text{Arc tg } t]$

Nous pouvons déterminer la constante de telle sorte que son erreur soit négligeable devant celle de l'approximation ; c'est pourquoi le tableau suivant ne tient compte que de la précision de $Ap. [\text{Arc tg } t]$

Nature de l'approximation	Ecart théorique fonction - approximation
Tchebycheff	$< 2,3 \cdot 10^{-10}$
Série-Relaxation	$< 1,12 \cdot 10^{-10}$
Rationnelle	$< 2,5 \cdot 10^{-11}$

Sur le nombre d'opérations

Nous ne tenons pas compte des opérations qui permettent de localiser l'argument dans un intervalle déterminé : en effet, elles sont nécessaires pour le choix, soit de la formule qu'il faut utiliser, soit des constantes qui doivent intervenir, mais elles ne concernent pas les calculs proprement dits.

Nous ne considérons donc que les opérations nécessaires au changement de variable et au calcul de l'approximation ; nous y ajoutons l'addition de la constante précalculée.

Nature de l'approximation	Additions	Multiplications	Divisions
Tchebycheff	6	5	1
Série-Relaxation	5	5	1
Rationnelle	7	2	3

Dans le cas particulier où $q = 0$, c'est-à-dire, $|x| < \operatorname{tg} \frac{\pi}{20}$, le nombre d'opérations pour les approximations, rationnelle et de Tchebycheff, est réduit de 3 additions et 1 division puisqu'il n'y a plus alors de changement de variable à effectuer et que la constante précalculée est nulle.

Pour le développement en série, nous avons 1 addition et 1 division supplémentaires lorsque l'argument est supérieur à l'unité.

Dans le cas courant, le tableau montre qu'en simple précision, le nombre d'opérations est sensiblement le même pour toutes les approximations, comme l'est aussi la précision. Par contre, en double précision, le développement en série avec relaxation est de beaucoup le plus avantageux.

122 - 5 Arc sinus

122 - 51 - Calcul numérique des approximations

Nous pouvons calculer Arc sinus par la relation :

$$\operatorname{Arc} \sin x = \operatorname{Arc} \operatorname{tg} \frac{x}{\sqrt{1-x^2}}$$

Les fonctions Racine carrée et Arc tangente ont été étudiées aux paragraphes 11 et 122 - 4.

Nous pouvons aussi obtenir Arc sinus par un calcul direct.

122 - 511 - Développement de Tchebycheff

Dans « Mathematical Methods for Digital Computers » (p.31), E.G. Kogbetliantz donne pour Arc sinus le développement suivant :

$$\operatorname{Arc} \sin (X \sin 2\varphi) = \sum_{h=0}^{\infty} C_h(\varphi) T_{2h+1}(X) = \sum_{h=0}^{\infty} a_{2h+1} T_{2h+1}(X)$$

$$\text{avec : } 0 \leq X \leq 1 ; 0 < \varphi < \frac{\pi}{4}$$

et la formule de récurrence :

$$h(2h+3)^2 C_{h+1} = (2h+1) [2h(h+1)(\operatorname{tg}^2 \varphi + \operatorname{cotg}^2 \varphi) + 1] C_h - (h+1)(2h-1)^2 C_{h-1}$$

Si nous limitons l'intervalle de variation à $\left[0, \frac{\sqrt{2}}{2}\right]$, φ est borné par $\frac{\pi}{8}$ et $\operatorname{tg}^2 \varphi + \operatorname{cotg}^2 \varphi$ égale 6.

REMARQUE : Pour $\frac{\sqrt{2}}{2} < X \sin 2\varphi \leq 1$, nous appliquons la formule :

$$\operatorname{Arc} \sin (X \sin 2\varphi) = \frac{\pi}{2} - \operatorname{Arc} \sin \sqrt{1 - X^2 \sin^2 2\varphi}$$

Avec $\varphi = \frac{\pi}{8}$, la formule de récurrence devient :

$$C_{h+1} = \frac{(2h+1)(12h^2 + 12h + 1)}{h(2h+3)^2} C_h - \frac{(h+1)(2h-1)^2}{h(2h+3)^2} C_{h-1}$$

$$C_0 = \frac{\sqrt{2}}{K} ; C_1 = C_0 - \frac{8K\sqrt{2}}{9\pi}$$

K représente l'intégrale elliptique complète de première espèce :

$$K = \int_0^{\pi/2} \frac{d\varphi}{\sqrt{1 - \sin^2 \alpha \sin^2 \varphi}} \quad \text{avec } \sin \alpha = \frac{\sqrt{2}}{2}$$

Calcul de K -

Nous avons calculé K directement à partir de l'intégrale, en développement en série $[1 - \sin^2 \alpha \sin^2 \varphi]^{-1/2}$. Nous obtenons après avoir intégré entre 0 et $\frac{\pi}{2}$:

$$K = \frac{\pi}{2} \left[1 + \left(\frac{1}{2}\right)^2 \sin^2 \alpha + \left(\frac{1}{2}\right)^2 \left(\frac{3}{4}\right)^2 \sin^4 \alpha + \dots \right] \quad \left(\sin^2 \alpha = \frac{1}{2}\right)$$

$$\text{ou } K = \frac{\pi}{2} \sum_{h=0}^{\infty} U_{2h} \quad \text{avec} \quad U_{2h} = \frac{1}{2} \left(\frac{2h-1}{2h}\right)^2 U_{2h-2} \quad \text{et } U_0 = 1$$

Le calcul de cette série, mené en triple précision (27 chiffres significatifs) et arrêté dès que le terme U_{2h} a une valeur inférieure à 10^{-27} , donne :

$$K = 1,854 \ 074 \ 677 \ 301 \ 371 \ 918 \ 433 \ 850 \ 38$$

Précision du résultat -

La série est très peu convergente; il faut donc calculer un grand nombre de termes avant d'atteindre : $U_{2h} = U_{2n} < 10^{-27}$. L'ordre n de troncature est très élevé et nous avons

alors U_{2n+2} peu différent mais inférieur à $\frac{1}{2} U_{2n}$

La somme des termes négligés admet pour borne supérieure :

$$\frac{\pi}{2} \sum_{h=n}^{\infty} U_{2h} < \frac{\pi}{2} U_{2n} \left[1 + \frac{1}{2} + \left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^3 + \dots \right] = \pi U_{2n}$$

c'est-à-dire que l'erreur de méthode est inférieure à $\pi \cdot 10^{-27}$

L'erreur sur le calcul de la somme des $(n-1)$ premiers termes est :

$$\Delta(K) < \frac{2}{\pi} K \Delta\left(\frac{\pi}{2}\right) + \frac{\pi}{2} (n-1) \Delta(U_{2h})$$

Nous avons mis le signe $<$ car nous nous sommes placés dans le cas défavorable où il y a cumul des erreurs d'arrondi de chaque terme U_{2h}

$$\Delta(U_{2h}) \leq 5 \cdot 10^{-28} \quad (U_{2h} < 1)$$

$$\Delta\left(\frac{\pi}{2}\right) \leq 5 \cdot 10^{-27} \quad \left(\frac{\pi}{2} > 1\right)$$

Nous déterminons une limite supérieure de "n" par l'inégalité :

$$U_{2n} = \left(\frac{1}{2}\right)^2 \left(\frac{3}{4}\right)^2 \dots \left(\frac{2n-1}{2n}\right)^2 \sin^{2n} \alpha < (\sin^2 \alpha)^n < 10^{-27}$$

Avec $\sin^2 \alpha = \frac{1}{2}$, nous avons $2^{-n} < 10^{-27}$ vérifié quand "n" est supérieur ou égal à 91.

$$\Delta(K) < \frac{2}{\pi} (1,855) (5 \cdot 10^{-27}) + \frac{\pi}{2} (90) (5 \cdot 10^{-28}) \simeq 76 \cdot 10^{-27}$$

L'erreur absolue sur le résultat est donc inférieure à : $(76 + \pi) \cdot 10^{-27} < 8 \cdot 10^{-26}$

Calcul des coefficients C_h ou a_{2h+1}

Avec la valeur de K trouvée, nous obtenons :

$$C_0 = 0,762 \ 759 \ 763 \ 501 \ 813 \ 188 \ 062 \ 325 \ 971$$

$$C_1 = 0,020 \ 869 \ 237 \ 569 \ 303 \ 689 \ 145 \ 499 \ 527$$

Les erreurs correspondantes sont : $\Delta(C_0) \leq 3,56 \cdot 10^{-26}$ et $\Delta(C_1) \leq 7,14 \cdot 10^{-26}$

Dès que h est assez grand, la formule de récurrence peut s'écrire : $C_{h+1} \sim 6 C_h - C_{h-1}$

L'erreur sur C_h est donc multipliée par 6 à chaque pas.

Si nous nous limitons à la simple précision pour le calcul de Arc sinus, le développe-

ment limité à l'ordre 13 est très largement suffisant .

Il suffit de déterminer une borne supérieure de l'erreur sur C_{14} ; chaque coefficient d'ordre inférieur à 15 est alors connu avec une erreur inférieure à cette borne .

$$\Delta (C_{14}) < 6^{13} \cdot \Delta (C_1) < 9,3 \cdot 10^{-16}$$

Ainsi l'erreur sur les 15 premiers termes donnés dans le tableau suivant, est inférieure à 10^{-15}

Valeurs des 15 premiers coefficients .

Nous avons programmé la formule de récurrence en triple précision afin de débiter le calcul avec des valeurs C_0 et C_1 aussi précises que possible. Toutefois, nous ne retenons que les 18 premières décimales, compte tenu de ce qui a été dit précédemment .

$C_0 = a_1$	=	0,762	759	763	501	813	188
$C_1 = a_3$	=	0,020	869	237	569	303	689
$C_2 = a_5$	=	0,001	586	931	627	766	012
$C_3 = a_7$	=	0,000	160	822	752	687	703
$C_4 = a_9$	=	0,000	018	691	074	902	953
$C_5 = a_{11}$	=	0,000	002	354	064	165	325
$C_6 = a_{13}$	=	0,000	000	312	575	618	546
$C_7 = a_{15}$	=	0,000	000	043	086	361	275
$C_8 = a_{17}$	=	0,000	000	006	107	059	218
$C_9 = a_{19}$	=	0,000	000	000	884	495	920
$C_{10} = a_{21}$	=	0,000	000	000	130	325	158
$C_{11} = a_{23}$	=	0,000	000	000	019	473	897
$C_{12} = a_{25}$	=	0,000	000	000	002	944	025†
$C_{13} = a_{27}$	=	0,000	000	000	000	449	474
$C_{14} = a_{29}$	=	0,000	000	000	000	069	195

Développement limité à l'ordre (n - 1) . Erreur de méthode .

Il suffit de choisir " n " de façon que l'inégalité suivante soit vérifiée :

$$\frac{C_n}{1 - \frac{C_{n+1}}{C_n}} < \varepsilon$$

1) Simple précision ($\varepsilon = 0,5 \cdot 10^{-9}$)

Le tableau précédent montre que :

$$\frac{C_{10}}{1 - \frac{C_{11}}{C_{10}}} < 1,6 \cdot 10^{-10} < \varepsilon$$

Le développement limité à l'ordre 9 fournit une approximation de Arc sinus :

$$Ap. \left[\text{Arc sin} \left(\frac{\sqrt{2}}{2} X \right) \right] = \sum_{h=0}^9 a_{2h+1} T_{2h+1}(X)$$

L'erreur de méthode correspondante est inférieure à $1,6 \cdot 10^{-10}$

2) Double précision ($\epsilon = 0,5 \cdot 10^{-18}$)

Le développement de Tchebycheff ne converge pas suffisamment pour poursuivre cette méthode en double précision. En effet, il faut déjà 10 termes en simple précision ; or, d'une manière générale, le nombre de termes double quand nous passons de la simple à la double précision.

En cas de nécessité, le calcul de Arc sinus en double précision se fera à l'aide de l'approximation de Arc tangente.

Formule numérique

$$Ap. \left[\text{Arc sin} \left(\frac{\sqrt{2}}{2} X \right) \right] = \sum_{n=0}^9 K_{2n+1} X^{2n+1}$$

avec $|X| \leq 1$

K_1	=	0,707	106	778	83
K_3	=	0,058	925	732	80
K_5	=	0,013	254	749	46
K_7	=	0,003	979	210	45
K_9	=	0,001	170	421	73
K_{11}	=	0,001	022	286	76
K_{13}	=	- 0,000	799	414	44
K_{15}	=	0,001	207	660	29
K_{17}	=	- 0,000	701	127	93
K_{19}	=	0,000	231	865	30

Ecart | fonction - approximation | $< 1,6 \cdot 10^{-10}$

122 - 512 - Développement d'Hastings

$$Ap. [\text{Arc sin } X] = \frac{\pi}{2} - \sqrt{1-X} \sum_{n=0}^7 K_n X^n$$

avec $0 \leq X \leq 1$

K_0	=	1,570	796	305	0
K_1	=	- 0,214	598	801	6
K_2	=	0,088	978	987	4
K_3	=	- 0,050	174	304	6
K_4	=	0,030	891	881	0
K_5	=	- 0,017	088	125	6
K_6	=	0,006	670	090	1
K_7	=	- 0,001	262	491	1

Ecart | fonction - approximation | $\leq 2 \cdot 10^{-8}$

122 - 52 - Comparaison entre le calcul direct de Arc sinus et le calcul par Arc tangente

Pour le développement de Tchebycheff, quand l'argument est compris entre 0 et $\frac{\sqrt{2}}{2}$ nous calculons Arc sinus à l'aide d'un développement de 10 termes ; mais quand il est compris entre $\frac{\sqrt{2}}{2}$ et 1, nous sommes obligés d'appliquer la formule :

$$\text{Arc sin } x = \frac{\pi}{2} - \text{Arc sin } \sqrt{1 - x^2}$$

c'est-à-dire de calculer en plus une racine.

Pour $\text{Arc sin } x = \text{Arc tg } \frac{x}{\sqrt{1-x^2}}$, nous avons toujours à effectuer le calcul d'une racine ; mais ensuite nous pouvons obtenir Arc tangente par un développement de 3 termes seulement.

Si nous adoptons, pour calculer la racine, la méthode exposée au paragraphe 111 - 2 et pour Arc tangente le développement en série (voir paragraphe 122 - 411 - 2), le nombre total d'opérations se répartit ainsi :

Arc sin x		Additions	Multiplications	Divisions
Calcul direct (Tchebycheff)	$0 \leq x \leq \frac{\sqrt{2}}{2}$	9	12	0
	$\frac{\sqrt{2}}{2} < x \leq 1$	15	14	2
Calcul par Arc tangente		10	8	4

Si l'argument est inférieur à $\frac{\sqrt{2}}{2}$ le calcul direct est plus avantageux, et, en outre, ne requiert aucune division; dans le cas contraire, le passage par Arc tangente est préférable. La précision n'influera pas puisqu'elle est sensiblement la même dans les deux procédés :

0,16 . 10^{-9} pour le premier

0,3 . 10^{-9} pour le second

SECONDE PARTIE



PROGRAMMATION SUR IBM 650 STANDARD

Les méthodes exposées dans la première partie ont été programmées en simple et double précisions sur l'ordinateur IBM 650 Standard.

21 - GENERALITES

211 - Référence des sous-programmes

Pour faciliter leur nomenclature, les sous-programmes seront référés dans la comparaison suivant la structure ci-dessous :

1	2	3	4	5	6
S	F	R	C	T	n
D		L	E	S	
		E	X	H	
		S	C	R	
		A	T		
		A	S		

Position 1 :

précision { S (simple)
 { D (double)

Position 2 :

F (programme fonctionnel)

Positions 3 et 4 :

nature de la fonction	{	RC	(racine carrée)
		LE	(logarithme à base e)
		EX	(exponentielle)
		SC	(sinus et cosinus)
		AT	(Arc tangente)
		AS	(Arc sinus)

Position 5 :

méthode employée	{	T	(Développement limité de Tchebycheff)
		S	(Développement en série et Relaxation)
		H	(Développement d'Hastings)
		R	(Approximation rationnelle)

Position 6 :

numéro de référence utilisé pour distinguer les différents développements possibles pour une même méthode. Par exemple, dans S F L E T 1, nous avons programmé le développement de Tchébycheff de $\text{Log}(1 + \lambda X)$ et dans S F L E T 2 celui de

$$\text{Log}\left(\frac{1 + \lambda X}{1 - \lambda X}\right)$$

C'est au cours de la comparaison que nous attribuerons éventuellement ce numéro.

Après avoir comparé les sous-programmes, nous retiendrons les meilleurs pour les inclure dans les bibliothèques. A précision sensiblement égale, l'encombrement et le temps de calcul guideront le choix. Il sera généralement nécessaire de disposer de deux sous-programmes : l'un, rapide, mais occupant davantage de mémoires ; l'autre, moins rapide, mais plus condensé. Nous ne les distinguerons donc plus par la méthode utilisée, mais par les deux critères mentionnés ci-dessus.

1	2	3	4	5
INCHANGE				K
				E
				(blanc)

Sous-programme : Temps de calcul réduit

" : Encombrement restreint

" : offrant les deux qualités

212 - Programmation

212 - 1 Simple précision

Nous avons adopté la programmation en virgule fixe plutôt qu'en virgule flottante puisque l'exécution des calculs est ainsi plus rapide. Nous pouvons, en outre, espérer une meilleure précision : en effet, les nombres peuvent avoir dix chiffres significatifs au lieu de huit en virgule flottante. Enfin, ce choix permet l'autonomie des sous-programmes fonctionnels, alors que la programmation en virgule flottante fait appel au programme AVF (arithmétique en virgule flottante).

Toutefois, lorsque les sous-programmes sont introduits dans des programmes généraux en virgule flottante du type FLAIR (exposant à gauche), nous adoptons la même représentation en opérant de la façon suivante : quelques instructions supplémentaires transforment l'écriture « virgule flottante » de l'argument en écriture « virgule fixe », renvoient au programme en virgule fixe pour le calcul de la fonction, puis, transforment l'écriture « virgule fixe » du résultat en écriture « virgule flottante ». Nous procédons ainsi afin de ne pas perdre le bénéfice de la virgule fixe. Dans ce cas, l'argument initial et le résultat final sont contenus exclusivement dans l'accumulateur gauche.

En virgule fixe, pour tous les sous-programmes, le cadrage choisi est 11 à partir de la position 20 des accumulateurs. En général, l'argument ne doit pas occuper plus de 10 positions. Nous verrons plus loin que pour certaines fonctions, il est possible de traiter des arguments de plus de 10 chiffres significatifs.

Le cadrage est le même pour l'entrée et la sortie du sous-programme, afin de permettre, le cas échéant, d'enchaîner directement un autre sous-programme fonctionnel. Cette remarque appelle certaines restrictions quand la fonction initiale est l'exponentielle ou la racine carrée : en effet, le résultat obtenu dans l'accumulateur doit être préalablement multiplié par la puissance de 10 contenue dans le distributeur.

Le cadrage 11 permet dans la plupart des cas (nombres compris entre -10 et 10) de n'utiliser qu'un seul accumulateur : pour entrer dans le sous-programme et en sortir, il suffit alors de deux instructions, l'une pour charger l'argument dans cet accumulateur, l'autre pour transférer le résultat dans une mémoire. Toutefois, il est toujours possible quand la valeur absolue du nombre est supérieure à 10, de faire appel au second accumulateur, ce qui double alors les instructions d'entrée et de sortie, mais étend l'intervalle jusqu'aux limites -10^{10} et 10^{10} . Enfin, avec ce cadrage, les nombres manipulés peuvent avoir 9 décimales.

Ce cadrage étant adopté, l'erreur de méthode doit porter sur la 10^e décimale. Il suffit alors de vérifier :

$$\text{Ecart} \mid \text{fonction} - \text{approximation} \mid \leq \varepsilon = 0,5 \cdot 10^{-9}$$

C'est cette valeur de ε qui avait été choisie au paragraphe 122.

212 - 2 Double précision

Les sous-programmes fonctionnels seront inclus dans des programmes généraux qui, en double précision, sont toujours établis en virgule flottante ; c'est pourquoi nous devons adopter le même mode de programmation pour faciliter l'enchaînement entre les sous-programmes. Ceci nous permet d'utiliser systématiquement pour tous les calculs, l'ADP (Arithmétique double précision, virgule flottante) qui a déjà été introduite pour les programmes généraux.

Nous pouvons également effectuer les mêmes calculs en virgule fixe, en ne conservant la virgule flottante que pour lier le sous-programme fonctionnel aux autres. Cela ne présente pas de difficultés puisque nous commençons toujours par réduire l'argument dans un intervalle donné, et que, par là, nous pouvons en déduire l'ordre de grandeur du résultat.

Ce second procédé permet de réduire d'environ cinq fois le temps de calcul obtenu avec le premier. Par contre, l'encombrement augmente de soixante mémoires environ : en effet, toutes les opérations arithmétiques en double précision, qui étaient effectuées par l'ADP dans le premier procédé, sont désormais à la charge du sous-programme.

Afin de réduire l'encombrement, tout en conservant ce temps de calcul avantageux, nous avons adopté le compromis suivant : nous utilisons l'ADP pour effectuer les multiplications et les divisions, opérations qui nécessitent un grand nombre d'instructions (respectivement 14 et 42 en virgule fixe et double précision). Pour toutes les autres (addition, remise à zéro, transfert), nous les laissons à la charge du sous-programme, parce qu'elles requièrent chacune 2 instructions et que le passage par l'ADP en demanderait autant. L'utilisation de l'ADP nous permet de gagner ainsi environ cinquante mémoires. Pour maintenir le gain du temps, nous gardons la programmation en virgule fixe, et, pour

celà, nous n'utilisons dans l'ADP, que les instructions opérant les calculs sur les mantisses.

En outre, nous pouvons, grâce à cette programmation, récupérer les deux positions qu'occuperait l'exposant et obtenir ainsi une meilleure précision.

A titre d'exemple, nous donnons l'encombrement (E) et le temps de calcul (K) obtenus pour les quatre fonctions les plus utilisées, programmées, d'une part (1), en effectuant tous les calculs sur des nombres flottants, d'autre part (2), en procédant comme nous venons de l'expliquer :

	$\sqrt{x}$		Sin x - Cos x		e^x		$\text{Log } e^x$	
	E	K	E	K	E	K	E	K
(1)	47	8,8sec	210	6 sec	163	8,8sec	178	9,5sec
(2)	64	2,3sec	197	1,45sec	158	1,95sec	184	2,1sec

Pour un encombrement sensiblement identique, le Temps de calcul est réduit d'environ cinq fois.

Puisque, comme nous venons de le voir, nous programmons en virgule fixe, le résultat est obtenu sous la même forme avant d'être transformé en écriture « virgule flottante » à l'aide de l'ADP. Pour entrer dans l'ADP, le cadrage, pour satisfaire certains impératifs propres à ce programme, doit être obligatoirement 2. Le résultat en virgule fixe a donc 18 décimales, et c'est pourquoi nous choisissons :

$$\text{écart} \mid \text{fonction} - \text{approximation} \mid \leq \varepsilon = 0,5 \cdot 10^{-18}$$

(valeur de ε mentionnée au paragraphe 122)

De cette façon, l'erreur de méthode porte sur la 19^e décimale.

REMARQUE : après que l'écriture « virgule fixe » du résultat ait été transformée en écriture « virgule flottante », la mantisse est composée des 18 premiers chiffres significatifs. Si la partie entière du résultat en virgule fixe est nulle et que la partie décimale possède « n » zéros en tête, l'erreur porte sur le $(19 - n)^e$ chiffre de la mantisse. Cet inconvénient ne peut être évité, car, pour que l'erreur porte sur le 19^e chiffre significatif, il faudrait alors enregistrer les coefficients des approximations avec $19 + n$ décimales : nous serions donc obligés d'utiliser une arithmétique de précision supérieure.

213 - Résultats de programmation

Pour chaque sous-programme d'une même fonction, nous notons l'encombrement, le temps de calcul et la précision obtenue. Ces données détermineront le choix des meilleurs sous-programmes.

213 - 1 Encombrement

L'encombrement E indique le nombre maximum de mémoires occupées par le sous-programme proprement dit. Les compléments annexes, tels que les transformations d'écriture en simple précision et en virgule flottante (voir paragraphe 212-1) et que l'ADP en double précision (voir paragraphe 212-2) requièrent aussi un certain nombre de mémoires. Mais ce nombre ne figure pas dans E car il reste constant, quelles que soient les méthodes envisagées, et ne modifie donc en rien la comparaison sur l'encombrement.

213 - 2 Temps de calcul

Le temps de calcul K indiqué est un temps moyen. Il est exprimé en millisecondes. Les mesures ont été effectuées pour de nombreuses valeurs de l'argument, échelonnées tout au long de l'intervalle de variation et nous retenons la moyenne arrondie de ces mesures.

213 - 3 Précision

La précision est donnée par « η » qui représente l'erreur maximale absolue sur les résultats ; « η » englobe :

- l'erreur de méthode évaluée dans la première partie,
- les erreurs dûes aux calculs (principalement les erreurs d'arrondi qui dépendent de la capacité du matériel),
- les erreurs inhérentes aux constantes précalculées.

Nous avons vu (paragraphe 212-1 et 212-2) que, quelle que soit la virgule utilisée dans les sous-programmes fonctionnels, nous obtenons toujours préalablement le résultat en virgule fixe. C'est pourquoi, pour la comparaison, nous relevons l'erreur sur ce résultat. Nous évitons ainsi l'inconvénient exposé dans la remarque du paragraphe 212-2.

REMARQUE : dans les notices d'utilisation des sous-programmes retenus, l'erreur indiquée porte sur la mantisse chaque fois qu'il s'agit de la virgule flottante.

En simple précision, nous avons déterminé facilement l'erreur par comparaison avec la double précision ou avec des tables existantes. En double précision, comme nous ne disposons pas de tables suffisamment précises, nous avons été obligés d'établir des programmes en triple précision à l'aide des développements en série (s'ils étaient suffisamment convergents) ou bien des développements de Tchébycheff donnés par M. VIONNET (Voir Annexe II (6)).

22 - COMPARAISON DES DIFFERENTES METHODES PROGRAMMEES

En vue de rendre la comparaison plus claire, nous la présentons sous forme de tableaux. Nous mentionnons dans chacun d'eux l'encombrement, le temps de calcul et la précision obtenus pour chaque sous-programme ainsi que, préalablement, les principaux facteurs qui influencent ces résultats. Dans une dernière colonne, nous désignons les meilleurs sous-programmes par la référence définitive qu'ils auront dans les bibliothèques (les raisons du choix sont encadrées).

REMARQUE 1 : afin de faciliter la comparaison des approximations polynomiales et rationnelles, qui se présentent sous des formes différentes, nous mentionnons pour chacune d'elles, le nombre d'opérations requises pour leur calcul. En double précision, nous ne tenons compte dans le tableau que des multiplications et des divisions ; elles exigent, en effet, respectivement 14 et 42 instructions, alors que l'addition n'en demande que 2 : son temps d'exécution, inférieur à 1,6 millisecondes, est donc négligeable devant le leur.

REMARQUE 2 : pour déterminer E , K , η , nous tenons compte de l'intervalle total de variation de l'argument et pas seulement de l'intervalle réduit car si le premier reste le même pour une fonction, quelle que soit l'approximation choisie, il n'en est pas forcément de même pour le second.

Après la comparaison, il nous a semblé bon d'expliquer l'écart obtenu entre la précision mentionnée dans le tableau, et celle que nous avons trouvée, pour chaque approximation, dans la première partie.

En effet, l'exécution des calculs peut entraîner trois sortes d'erreurs :

- l'« erreur de chute » après l'exécution d'une multiplication ou d'une division. Le nombre de décimales conservées est fini ; il dépend du nombre de positions disponibles (en général, 10 en simple précision, 20 en double précision) et du cadrage du résultat ,
- l'« erreur due aux données » : elle résulte de l'imprécision des constantes (coefficients des approximations) ou des résultats partiels préalablement calculés qui interviennent dans les opérations,
- l'« erreur d'arrondi classique » chaque fois qu'il y a perte de chiffres significatifs après un cadrage (les arrondis sont effectués sans tenir compte de la règle de parité) .

Nous avons étudié, pour les sous-programmes retenus, la propagation de toutes ces erreurs au cours du calcul ; elle conduit à une erreur finale sur le résultat qui, en général, est très supérieure à l'erreur de méthode évaluée dans la première partie .

C'est ce que nous entendons par « Perte de précision » .

221 - Racine Carrée

221 - 1 Simple précision

TABLEAU DE COMPARAISON

Référence	Intervalle réduit	Première approximation	Obtention de la racine	C ^{tes}	E	K	η	Référence dans la bibliothèque
SFRC 1	$(10^{-2}, 1)$	1	Calcul itératif arrêt pour $\frac{a}{x_n} \geq x_n$	4	49	365	$5 \cdot 10^{-10}$	SFRC E
SFRC 2	$(1, 10)$	$\sqrt{s+1}$	Calcul itératif arrêt pour $\frac{a}{x_n} \geq x_n$	11	61	260	$5 \cdot 10^{-10}$	
SFRC 3	$(10^{-1}, 1)$	$A(B + a \frac{C}{D+a})$	$\frac{1}{2}(\frac{a}{x_0} + x_0)$	21	105	160	$2 \cdot 10^{-9}$	SFRC K

SFRC 1 - La première approximation est $\frac{1}{2}(a+1)$. Nous pouvons aussi bien partir de $x_0 = 1$ car $x_1 = \frac{1}{2}(a+1)$: il en résulte un léger gain sur l'encombrement.

SFRC 2 - La première approximation est $\frac{1}{2}(\frac{a}{\sqrt{s}} + \sqrt{s})$: s est la partie entière de a (voir paragraphe 111 - 1). Nous ne pouvons pas partir de $x_0 = \sqrt{s}$ car $\frac{a}{\sqrt{s}}$ est supérieur ou égal à $\sqrt{s}$, inégalité qui correspond au test d'arrêt des itérations.
Pour éviter de faire le calcul de la première approximation séparément, il est plus avantageux de partir de $x_0 = \sqrt{s+1}$ et de passer directement aux itérations.

Le calcul itératif permet d'obtenir pour SFRC 1 et SFRC 2 un encombrement plus faible. Néanmoins le gain est moindre pour SFRC 2 parce qu'il faut rechercher la valeur de $\sqrt{s+1}$ dans une table précalculée, qui, de plus, occupe 10 mémoires.

Le temps de calcul varie en sens inverse de l'encombrement puisqu'un plus grand nombre d'instructions offre au départ une meilleure approximation de la racine. Pour SFRC 1 et SFRC 2, ce temps de calcul varie avec le nombre d'itérations ; il est constant pour SFRC 3.

Précision

Nous disposons en simple précision de 10 positions. Pour SFRC 1 et SFRC 2, il suffit d'arrêter le

calcul itératif dès que x_n et x_{n+1} ont 10 chiffres communs : l'erreur porte alors sur le 11^e chiffre significatif, c'est-à-dire, la 10^eème décimale puisque le résultat est cadré 1. La précision ne dépend donc alors que de la capacité du matériel. Par contre, pour SFRC 3, nous avons vu en fin du paragraphe 111 - 2 que l'erreur de méthode porte toujours sur le 10^e chiffre significatif même si la capacité est plus importante.

221 - 2 Double précision

TABLEAU DE COMPARAISON

Référence	Intervalle réduit	Première approximation	Obtention de la racine	C^{tes}	E	K	η	Référence dans la bibliothèque
DFRC 1	$(10^{-2}, 1)$	1	Calcul itératif. Arrêt dès que $\frac{a}{x_n}$ et x_n ont 18 décimales communes	3	64	2300	1.10^{-18}	DFRCE
DFRC 10	$(10^{-2}, 1)$	La racine calculée en simple précision à partir de 1	id	4	79	850	1.10^{-18}	
DFRC 2	$(1, 10)$	$\sqrt{s+1}$	id	12	82	1840	5.10^{-19}	
DFRC 20	$(1, 10)$	La racine calculée en simple précision à partir de $\sqrt{s+1}$	id	12	95	700	5.10^{-19}	DFRCK
DFRC 30	$(10^{-1}, 1)$	La racine calculée en simple précision à partir de : $A\left(B+a-\frac{C}{D+a}\right)$	id	18	120	860	1.10^{-18}	

Dans DFRC 10 - 20 - 30, nous prenons pour première approximation la racine calculée en simple précision, ce qui permet de réduire le nombre d'itérations. Il en résulte un encombrement plus important mais cet inconvénient est largement compensé par le gain de temps. Par exemple, pour DFRC 10 et DFRC 20 l'encombrement augmente respectivement de 23 et 16 % par rapport à celui de DFRC 1 et DFRC 2, mais le calcul est presque 3 fois plus rapide.

DFRC 30 ne donne pas les meilleurs temps bien que le calcul en simple précision soit de beaucoup le plus rapide (voir paragraphe 221 - 1). En effet, l'erreur relative à la première approximation porte sur le 10^e chiffre au lieu du 11^e dans les autres procédés ; c'est pourquoi il faut faire une itération de plus en double précision.

Précision

En double précision, nous arrêtons les itérations dès que $\frac{a}{x_n}$ et x_n ont 18 décimales communes. Pour DFRC 2 et DFRC 20, comme l'intervalle réduit est $(1, 10)$, le résultat a donc 19 chiffres significatifs. La mantisse est formée des 18 premiers : il en résulte une erreur d'arrondi de 5.10^{-19} .

Pour les autres sous-programmes, l'intervalle réduit est $(10^{-2}, 1)$ ou $(10^{-1}, 1)$. Le résultat, inférieur à 1, n'a que 18 chiffres significatifs : l'erreur sur la mantisse est l'« erreur de chute » soit 1.10^{-18} .

222 - Logarithme népérien

222 - 1 Simple précision

TABLEAU DE COMPARAISON

Référence	Intervalle réduit	Changement de variable	Approximation		Ctes	E	K	η	Référence dans la bibliothèque
SFLET 1	$(10^{-\frac{1}{16}}, 10^{\frac{1}{16}})$	$X = \frac{1}{\lambda}(x - 1)$	7 termes d'ordre consécutif	7 additions 7 multipl.	19	103	275	2.10^{-9}	
SFLET 2	$(10^{-\frac{1}{16}}, 10^{\frac{1}{16}})$	$X = \frac{1}{\lambda} \frac{x-1}{x+1}$	3 termes d'ordre impair	2 additions 4 multipl.	15	94	270	1.10^{-9}	
SFLEH1	$(1, 10)$	$t = \frac{x - \sqrt{10}}{x + \sqrt{10}}$	5 termes d'ordre impair	4 additions 6 multipl.	11	64	180	3.10^{-7}	
SFLEH2	$(1, 10^{\frac{1}{4}})$	$t = x - 1$	8 termes d'ordre consécutif	7 additions 8 multipl.	15	90	270	3.10^{-8}	
SFLES	$(10^{-\frac{1}{16}}, 10^{\frac{1}{16}})$	$t = \frac{x-1}{x+1}$	3 termes d'ordre impair	2 additions 4 multipl.	14	91	250	2.10^{-9}	SFL E
SFLER	$(10^{-\frac{1}{16}}, 10^{\frac{1}{16}})$	$t = \frac{x-1}{x+1}$	expression rationnelle	2 additions 2 multipl. 2 divisions	14	94	270	1.10^{-9}	

REMARQUE : Pour tous les sous-programmes, l'intervalle de variation de l'argument est $(10^{-9}, 10^{10})$.

Le résultat final est : $k \text{ LOG}_e 10 + \text{Approximation}$.

La valeur de k est fonction de la réduction d'intervalle. (Voir en Annexe I le détail de SFL E).

D'après ce tableau comparatif, nous voyons qu'aucune méthode ne l'emporte nettement sur les autres. Nous ne retenons pas les développements d'Hastings, puisque la précision est plus faible d'une ou deux décimales, et que, par là, nous ne pouvons pas tenir compte de leur encombrement et de leur temps de calcul, même inférieurs.

SFLET 1 (développement de $\text{Log}(1 + \lambda X)$) occupe environ dix mémoires de plus pour ne présenter, par ailleurs, aucun avantage.

Les autres développements, issus de $\text{Log} \frac{1 + \lambda X}{1 - \lambda X}$ convergent plus rapidement (3 termes au lieu de 7 pour les approximations polynomiales) mais les gains obtenus sur l'encombrement et le temps, dans le calcul de l'approximation, sont, en partie, perdus par le changement de variable.

Perte de précision

Nous comparons dans ce tableau, l'écart théorique entre la fonction et l'approximation utilisée (erreur de méthode) et l'écart réel entre la fonction et le résultat (η).

Référence	Erreur de méthode	η
SFLET 1	5.10^{-10}	2.10^{-9}
SFLET 2	5.10^{-11}	1.10^{-9}
SFLES	6.10^{-11}	2.10^{-9}
SFLEH 1	$\approx 10^{-7}$	3.10^{-7}
SFLEH 2	3.10^{-8}	3.10^{-8}
SFLER	5.10^{-10}	1.10^{-9}

Nous voyons, d'après ce tableau, que, sauf pour les développements d'Hastings, l'erreur de méthode est pratiquement négligeable devant l'erreur due aux calculs. Cette dernière résulte de la propagation des erreurs définies au début du paragraphe 22, et nous allons montrer pourquoi elle est si importante. Notre étude porte sur SFLES (Voir en Annexe 1 le détail de ce sous-programme).

Au cours de la réduction d'intervalle dans $(10^{-\frac{1}{16}}, 10^{\frac{1}{16}})$ l'argument reste cadré 1, car sa valeur peut être supérieure à l'unité. Avec 10 positions, l'erreur de chute est inférieure à 1.10^{-9} (Nous ne tenons pas compte ici de l'erreur due aux données, qui reste négligeable devant 10^{-9}).

Le changement de variable $t = \frac{x-1}{x+1}$ donne une valeur inférieure à 1 et nous pouvons ainsi choisir pour t le cadrage 0. Après la division, l'erreur de chute est 1.10^{-10} ; l'erreur due à l'imprécision de x est $\frac{2\Delta(x)}{(x+1)^2}$ soit $0,7 \cdot 10^{-9}$: nous connaissons donc t avec une erreur $\Delta(t)$ inférieure à 8.10^{-10} . L'approximation $A = K_1 t + K_3 t^3 + K_5 t^5$ conduit à un résultat inférieur à 1: il peut

être cadré 0. L'erreur sur A est : $\Delta(A) = t \Delta(K_1) + t^3 \Delta(K_3) + t^5 \Delta(K_5) + \Delta(t) [K_1 + 3t^2 K_3 + 5t^4 K_5]$

En tenant compte des erreurs dues aux données et de l'erreur de chute, nous obtenons

$$\Delta(A) \leq 1,8 \cdot 10^{-9}.$$

L'erreur sur $k \log_e 10$ est négligeable devant $1,8 \cdot 10^{-9}$.

L'erreur due aux calculs est donc de l'ordre de $2 \cdot 10^{-9}$.

222 - 2 Double précision

TABLEAU DE COMPARAISON

Référence	Intervalle réduit	Changement de variable	Approximation		C ^{tes}	E	K	η	Référence dans la bibliothèque
DFLET 1	$\left(10^{-\frac{1}{16}}, 10^{\frac{1}{16}}\right)$	$X = \frac{1}{\lambda}(x-1)$	15 termes d'ordre consécutif	15 multipl.	31	184	2100	$5 \cdot 10^{-18}$	
DFLET 2	id	$X = \frac{1}{\lambda} \frac{x-1}{x+1}$	6 termes d'ordre impair	7 multipl.	24	156	1600	$5 \cdot 10^{-18}$	
DFLES	id	$t = \frac{x-1}{x+1}$	6 termes d'ordre impair	7 multipl.	23	154	1400	$5 \cdot 10^{-18}$	DFLE
DFLER	id	$t = \frac{x-1}{x+1}$	expression rationnelle	7 multipl. 1 division	24	154	1650	$6 \cdot 10^{-18}$	

REMARQUE : Pour tous les sous-programmes, l'intervalle de variation de l'argument est $(1 \cdot 10^{-42}, 1 \cdot 10^{42})$

Le résultat final est : $k \log_e 10 + \text{Approximation}$.

La valeur de k est fonction de la réduction d'intervalle.

Alors qu'en simple précision, la différence entre les développements de $\log(1+\lambda X)$ et de $\log \frac{1+\lambda X}{1-\lambda X}$ était peu sensible, elle apparaît, au contraire, très marquée en double précision. En effet, dans DFLET 1, nous avons un développement de 15 termes au lieu de 6 ou 7 dans les autres, pour obtenir la même précision; il en résulte, par rapport à DFLES, une augmentation de 18 % sur l'encombrement, et surtout de 50 % sur le temps de calcul.

Perte de précision

Référence	Erreur de méthode	η
SFEX.T	8.10^{-11}	3.10^{-9}
SFEX S	5.10^{-10}	6.10^{-9}
SFEX H	5.10^{-8}	5.10^{-8}
SFEX R 1	8.10^{-12}	3.10^{-9}
SFEX R 2	6.10^{-13}	6.10^{-9}
SFEX R 3	9.10^{-13}	7.10^{-9}
SFEX R 4	2.10^{-11}	7.10^{-9}

Pour le développement d'Hastings, l'erreur due aux calculs est négligeable devant l'erreur de méthode ; il se produit l'inverse pour les autres approximations .

En tenant compte des erreurs de chute et des erreurs dues aux données, nous avons trouvé, sur le calcul de l'approximation, une erreur inférieure à $1,1 \cdot 10^{-9}$ dans le cas le plus défavorable. (Voir en Annexe I le détail des sous-programmes SFEXK et SFEXE). Le résultat final est : $10^m \times$ Approximation . L'erreur sur ce produit s'écrit :

$$10^m \cdot \Delta(A) + A \cdot \Delta(10^m) \text{ avec } \Delta(A) = 1,1 \cdot 10^{-9}$$

L'erreur sur 10^m est inférieure à 4.10^{-10} , quelle que soit la valeur de m compris entre 0 et 1 . La valeur maximale de A est 10^{1-m} puisque le résultat final est ramené dans l'intervalle (1,10) .

$$\text{L'erreur devient : } \left(1,1 \cdot 10^m + \frac{4}{10^m} \right) 10^{-9}$$

Nous en déduisons que :

- l'erreur varie avec la valeur de m et dépend donc de l'intervalle réduit ,
- l'erreur porte au plus sur le chiffre de rang 10 (ou sur la neuvième décimale) , quelle que soit la valeur de m .

En effet $y = \left(1,1 \cdot 10^m + \frac{4}{10^m} \right) 10^{-9}$ est d'abord décroissant, passe par un minimum pour

$1,1 \cdot 10^m = \frac{4}{10^m}$ puis croit . La valeur maximale de y est atteinte dans le programme SFEX R 4

pour $m = \frac{15}{16}$ soit $y_{\max} \simeq 9,96$.

223 - 2 Double précision

TABLEAU DE COMPARAISON

Référence	Intervalle réduit	Approximation		C ^{tes}	E	K	η	Référence dans la bibliothèque
DFEXT	$(-\frac{L}{4}, \frac{L}{4})$	15 termes	14 multiplic.	24	158	1950	5.10^{-18}	
DFEXS	$(-\frac{L}{8}, \frac{L}{8})$	12 termes	11 multiplic.	28	163	1700	5.10^{-18}	DFEX
DFEXR1	$(-\frac{L}{2}, \frac{L}{2})$	2 fois 10 termes	18 multiplic 1 division	18	150	2870	3.10^{-18}	
DFEXR2	$(-\frac{L}{4}, \frac{L}{4})$	2 fois 8 termes	14 multiplic 1 division	18	148	2400	3.10^{-18}	
DFEXR3	$(-\frac{L}{8}, \frac{L}{8})$	2 fois 7 termes	12 multiplic 1 division	23	163	2200	3.10^{-18}	
DFEXR4	$(-\frac{L}{16}, \frac{L}{16})$	2 fois 6 termes	10 multiplic 1 division	28	175	2050	3.10^{-18}	

$L = \text{Log}_e 10$

REMARQUE : Pour tous les sous-programmes, l'intervalle de variation de l'argument est $(-110, 110)$

Le résultat final est : 10^m . Approximation (Voir les Remarques du paragraphe 223 - 1)

L'encombrement augmente de 16 % quand q passe de 1 à 4 (32 % en simple précision). Le temps de calcul varie sensiblement dans les mêmes proportions qu'en simple précision (voir paragraphe 223 - 1).

DFEXS donne le meilleur temps, DFEXR2 le plus faible encombrement. Toutefois, le gain sur l'encombrement de DFEXR2 sur DFEXS n'est que de 9 % alors que la perte sur le temps de calcul est de 41 % ; il n'est donc pas suffisamment important pour retenir deux sous-programmes. C'est pourquoi nous gardons seulement DFEXS.

D'après le tableau ci-après, nous constatons que l'erreur de méthode est négligeable devant l'erreur due aux calculs. Une rapide étude permet de déterminer l'ordre de grandeur de cette dernière erreur.

Perte de précision

Référence	Erreur de méthode	η
DFEX T	$1.2 \cdot 10^{-20}$	$5 \cdot 10^{-17}$
DFEX S	$8 \cdot 10^{-19}$	$5 \cdot 10^{-17}$
DFEX R1	$2 \cdot 10^{-20}$	$3 \cdot 10^{-17}$
DFEX R2	$2 \cdot 10^{-19}$	$3 \cdot 10^{-17}$
DFEX R3	$3 \cdot 10^{-20}$	$3 \cdot 10^{-17}$
DFEX R4	$7 \cdot 10^{-20}$	$3 \cdot 10^{-17}$

En effet, l'argument limité à 110 est d'abord réduit dans l'intervalle $(-\text{Log}_e 10, \text{Log}_e 10)$ en posant :

$$x \cdot \text{Log}_{10} e = N + F$$

La partie entière N peut varier entre -48 et 48 : nous devons donc cadrer 2 le résultat de la multiplication. En tenant compte des 20 positions, nous avons une erreur de chute de 10^{-18} sur la partie décimale F . Cette erreur reste inchangée dans la poursuite de la réduction.

L'erreur sur le calcul de l'approximation est :

$$\Delta(K_0) + f \Delta(K_1) + \dots + \Delta(f) [K_1 + 2fK_2 + \dots]$$

En tenant compte des erreurs dues aux données, nous avons trouvé, dans le cas du développement en série avec relaxation, une erreur de $3,5 \cdot 10^{-18}$.

Dans le cas défavorable, il convient d'ajouter à cette erreur $11 \cdot 10^{-19}$ qui est la somme des erreurs de chute à la suite des 11 multiplications.

Nous connaissons alors l'approximation avec une erreur inférieure à $4,6 \cdot 10^{-18}$.

L'erreur sur le résultat final est :

$$10^m \cdot \Delta(A) + A \Delta(10^m) + \text{erreur de chute}$$

mais les deux derniers termes sont négligeables devant le premier.

Pour $m = \frac{7}{8}$, l'erreur est donc de l'ordre de :

$$10^{7/8} \cdot (4,6) \cdot 10^{-18} \simeq 3,5 \cdot 10^{-17}$$

224 - Sinus - Cosinus

224 - 1 Simple précision

TABLEAU DE COMPARAISON

Référence	Intervalle réduit	Approximation		C ^{tes}	E	K	η	Référence dans la bibliothèque
SFSC T1	$(-\frac{\pi}{4}, \frac{\pi}{4})$	5 termes d'ordre impair ou d'ordre pair	4 additions 6 multipl. 4 additions 5 multipl.	15	87	170	1.10^{-9}	
SFSC T2	$(-\frac{\pi}{2}, \frac{\pi}{2})$	6 termes d'ordre impair	5 additions 7 multipl.	9	57	170	1.10^{-9}	
SFSC T3	$(-\frac{\pi}{2}, \frac{\pi}{2})$	6 termes d'ordre pair	5 additions 6 multipl.	9	53	155	2.10^{-9}	SFSC
SFSC S	$(-\frac{\pi}{4}, \frac{\pi}{4})$	5 termes d'ordre impair ou d'ordre pair	4 additions 6 multipl. 4 additions 5 multipl.	14	85	160	2.10^{-9}	
SFSC H	$(-\frac{\pi}{2}, \frac{\pi}{2})$	5 termes d'ordre impair	4 additions 6 multipl.	8	53	155	6.10^{-9}	
SFSC R	$(-\frac{\pi}{4}, \frac{\pi}{4})$	forme rationnelle	7 additions 1 multipl. 5 divisions	13	106	230	1.10^{-9}	

REMARQUE : Pour tous les sous-programmes, l'intervalle de variation de l'argument est $(-10^{10}, 10^{10})$

Pour les approximations polynomiales, l'encombrement augmente de 54 % environ lorsque nous choisissons pour intervalle $[-\frac{\pi}{4}, \frac{\pi}{4}]$ au lieu de $[-\frac{\pi}{2}, \frac{\pi}{2}]$.

Cette augmentation ne résulte pas tant d'une réduction plus poussée que de la nécessité d'introduire deux développements pour couvrir tout l'intervalle de variation du résultat ; nous n'utilisons suivant la valeur de l'argument, que l'un ou l'autre des développements, c'est-à-dire, qu'une partie seulement des instructions sert. Le choix de cet intervalle permet de gagner un terme sur les développements, ce qui ne modifie pas sensiblement le temps en simple précision, compte tenu de celui qui est perdu pour choisir le développement adéquat.

L'approximation rationnelle paraît a priori désavantageuse, tant du point de vue de l'encombrement que du temps de calcul. En effet, pour obtenir les fonctions trigonométriques, il faut déterminer préalablement la tangente de l'arc moitié. Le calcul n'est donc pas direct, ce qui explique ce désavantage. Mais, contrairement aux autres approximations, il permet de calculer simultanément le sinus et le cosinus sans reprendre tout le programme : le calcul de la tangente de l'arc moitié, qui requiert le plus grand nombre d'instructions (5 additions - 1 multiplication - 3 divisions), se fait une seule fois pour servir ensuite aux calculs des deux fonctions qui ne demandent plus alors qu'une addition et une division, chacune.

Nous en déduisons que l'emploi de ce programme, pour le calcul de l'une ou l'autre fonction, est à proscrire. Il demande, en effet, par rapport à SFSC T3, un encombrement double et un temps de calcul supérieur.

S'il faut calculer à la fois les deux fonctions, nous réalisons une économie de 26 % sur le temps, puisque, dans un tel cas, nous devons utiliser 2 fois SFSC T3.

Néanmoins, comme la perte sur l'encombrement est très importante, il faut que l'approximation rationnelle soit utilisée pour un calcul répétitif et simultané de sinus et cosinus, afin de compenser cette perte par un gain de temps important. Par exemple, SFSCR est parfaitement adapté à la construction d'une table trigonométrique.

Perte de précision

Référence	Erreur de méthode	η
SFSC T1	5.10^{-11}	1.10^{-9}
SFSC T2	$1,4.10^{-11}$	1.10^{-9}
SFSC T3	$2,2.10^{-10}$	2.10^{-9}
SFSC S	2.10^{-10}	2.10^{-9}
SFSC H	5.10^{-9}	6.10^{-9}
SFSCR	$8,4.10^{-12}$	1.10^{-9}

Pour le développement d'Hastings, l'erreur de méthode est du même ordre de grandeur que l'erreur due aux calculs : il y a somme de ces deux erreurs.

Pour les autres approximations, l'erreur due aux calculs est beaucoup plus importante que l'erreur de méthode. La perte de précision provient surtout de la réduction d'intervalle. En effet, nous divisons tout d'abord l'argument par 2π . Comme le reste peut être supérieur à l'unité, nous le cadrons 1 : avec 10 positions, nous avons alors une erreur de chute inférieure à 1.10^{-9} . Dans la poursuite de la réduction, il peut y avoir cumul des erreurs sur π et $\frac{\pi}{2}$. Pour SFSC T3, l'erreur sur l'argument réduit dans $(-\frac{\pi}{2}, \frac{\pi}{2})$ est inférieure à $1,6.10^{-9}$ (Voir en Annexe I le détail de ce sous-programme).

L'erreur sur l'approximation est :

$$\Delta(A) = \Delta(K_0) + X^2 \Delta(K_2) + \dots + \Delta(X) [2 X K_2 + 4 X^3 K_4 + \dots]$$

En tenant compte de l'imprécision des coefficients K_{2n} et de $\Delta(X) < 1,6 \cdot 10^{-19}$, nous avons trouvé, dans le cas le plus défavorable, une erreur sur A inférieure à $5,4 \cdot 10^{-9}$.

224 - 2 Double précision

TABLEAU DE COMPARAISON

Référence	Intervalle réduit	Approximation		C^{tes}	E	K	η	Référence dans la bibliothèque
DFSCT1	$(-\frac{\pi}{4}, \frac{\pi}{4})$	8 termes d'ordre impair ou d'ordre pair	9 multipl. 8 multipl.	27	197	1450	$5 \cdot 10^{-18}$	
DFSCT2	$(-\frac{\pi}{2}, \frac{\pi}{2})$	9 termes d'ordre impair	10 multipl.	20	162	1650	$5 \cdot 10^{-18}$	
DFSCT3	$(-\frac{\pi}{2}, \frac{\pi}{2})$	10 termes d'ordre pair	10 multipl.	20	159	1650	$5 \cdot 10^{-18}$	DFSC
DFSCS	$(-\frac{\pi}{4}, \frac{\pi}{4})$	8 termes d'ordre impair ou d'ordre pair	9 multipl. 8 multipl.	26	200	1450	$5 \cdot 10^{-18}$	

REMARQUE : Pour tous les sous-programmes, l'intervalle de variation de l'argument est $(-10^3, 10^3)$.

En double précision, le choix de l'intervalle $(-\frac{\pi}{4}, \frac{\pi}{4})$ permet, en diminuant le nombre de termes de l'approximation, de réaliser une économie de 12 % sur le temps de calcul, avec, toutefois une augmentation de l'encombrement de 25 % (54 % en simple précision). Ces pourcentages ne sont pas suffisamment élevés pour retenir deux sous-programmes, l'un donnant le plus faible encombrement, l'autre le meilleur temps. Nous retenons donc DFSCT3 puisqu'il est le moins encombrant.

Perte de précision

L'erreur relevée sur le résultat en virgule fixe est inférieur à $5 \cdot 10^{-18}$ pour un argument initial inférieur à 10 en valeur absolue.

Référence	Erreur de méthode	η
DFSCT1	3.10^{-20}	5.10^{-18}
DFSCT2	$1,7.10^{-19}$	5.10^{-18}
DFSCT3	$6,4.10^{-21}$	5.10^{-18}
DFSCS	9.10^{-20}	5.10^{-18}

D'après ce tableau, nous voyons que l'erreur maximale η est la même, quelle que soit l'approximation envisagée. En effet, l'erreur de méthode n'influe que très peu sur η puisqu'elle est très inférieure ; nous allons montrer que η dépend essentiellement de la grandeur de l'argument initial.

Nous commençons par réduire l'argument dans l'intervalle $(-2\pi, 2\pi)$ en posant :
 $\frac{x}{2\pi} = N + F$. N est la partie entière, F la partie décimale.

Le cadrage de la virgule, et, par conséquent, l'erreur de chute sur F varie avec la valeur de l'argument. Si son exposant est inférieur ou égal à 50, l'erreur de chute sur F est 1.10^{-20} , s'il est supérieur, l'erreur est $10^{\alpha\alpha-70}$ (c'est pourquoi, dans les sous-programmes, l'exposant est limité à 53). Nous y ajoutons l'erreur due au calcul qui, suivant la valeur de l'argument, peut, elle aussi, être importante. Nous avons alors :

$$\text{erreur sur } F \leq \frac{x}{4\pi^2} \Delta(2\pi) + 10^{\alpha\alpha-70} \quad (\alpha\alpha > 50)$$

Afin de poursuivre l'étude, nous choisissons à titre d'exemple SFSCT3 et limitons l'argument entre -10 et 10.

$$\text{erreur sur } F \leq \frac{10}{4\pi^2} \Delta(2\pi) + 10^{-19}$$

2π est cadré 1 et limité à 18 chiffres significatifs (Voir en remarque la justification de cette limite). Son erreur est $\Delta(2\pi) = 3.10^{-18}$; l'erreur sur F est alors $\Delta(F) \leq 8,5.10^{-19}$. Cette erreur reste inchangée dans la poursuite de la réduction et le changement de variable.

L'erreur sur le calcul de l'approximation est :

$$\Delta(A) = \Delta(K_0) + \Delta(K_2) + \dots + \Delta(X) [2XK_2 + 4X^3K_4 + \dots] \quad (|X| \leq 1)$$

En tenant compte de la valeur des coefficients et de leur précision, ainsi que de $\Delta(X) \leq 8,5.10^{-19}$, nous obtenons $\Delta(A) \simeq 3,4.10^{-18}$. L'approximation est cadrée 2 (cadrage imposé pour l'entrée dans l'ADP), il faut donc ajouter à $3,4.10^{-18}$ l'erreur d'arrondi 5.10^{-19} . L'erreur sur le résultat final est de l'ordre de 4.10^{-18} quand l'argument a un exposant égal à 51 ; elle atteint $3,5.10^{-17}$ quand l'exposant égale 52.

REMARQUE : Nous enregistrons la constante 2π avec 18 chiffres significatifs au lieu de 20 comme le permettait la capacité. Nous pouvons ainsi obtenir des résultats exacts pour le sinus et le cosinus des arcs remarquables (forme $k \frac{\pi}{2}$). En effet, à ces arcs correspondent théoriquement les valeurs 0 et

1 pour la variable X, après réduction dans l'intervalle $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$. Or, ces valeurs doivent être exactes pour que nous puissions les tester et éviter ainsi le calcul par le développement qui donnerait un résultat approché à cause des erreurs inhérentes à l'approximation. Ceci n'est possible que si 2π a le même nombre de chiffres significatifs que $k\frac{\pi}{2}$, c'est-à-dire 18 (La valeur de $k\frac{\pi}{2}$ est donnée en virgule flottante). Il en résulte naturellement une plus grande erreur pour les arcs quelconques. Le calcul de la précision, repris dans les mêmes conditions que précédemment, mais en introduisant 2π avec 20 chiffres significatifs, donne une erreur de $1,2 \cdot 10^{-18}$ au lieu de $4 \cdot 10^{-18}$. L'erreur est, certes, plus faible, mais porte tout de même sur la 18^e décimale.

225 - Arc Tangente

225 - 1 Simple précision

TABLEAU DE COMPARAISON

Référence	Intervalle réduit	Changement de variable	Approximation		C ^{tes}	E	K	η	Référence dans la biblioth.
SFATT1	$\left(-\operatorname{tg} \frac{\pi}{16}, \operatorname{tg} \frac{\pi}{16}\right)$	$t = a - \frac{b}{q} \frac{q}{ x + a}$ $q = 0, 1, 2, 3, 4$	4 termes	3 additions 5 multipl.	20	102	185	$1 \cdot 10^{-9}$	
SFATT2	$\left(\operatorname{tg} \frac{\pi}{16}, \operatorname{tg} \frac{\pi}{16}\right)$	$t = a - \frac{b}{q} \frac{q}{ x + a}$ $q = 0, 1, 2$	4 termes	3 additions 5 multipl.	16	93	195	$2 \cdot 10^{-9}$	
SFATS	$(-1, 1)$	$t = \frac{x - x_0}{1 + x x_0}$	3 termes	2 additions 4 multipl.	17	86	175	$2 \cdot 10^{-9}$	SFAS
SFATH	$(-1, 1)$	néant	8 termes	7 additions 9 multipl.	12	71	205	$4 \cdot 10^{-8}$	
SFATR1	$\left(-\operatorname{tg} \frac{\pi}{16}, \operatorname{tg} \frac{\pi}{16}\right)$	$t = a - \frac{b}{q} \frac{q}{ x + a}$ $q = 0, 1, 2, 3, 4$	forme rationnelle	4 additions 2 multipl. 2 divisions	21	103	190	$1, 10^{-9}$	
SFATR2	$\left(-\operatorname{tg} \frac{\pi}{16}, \operatorname{tg} \frac{\pi}{16}\right)$	$t = a - \frac{b}{q} \frac{q}{ x + a}$ $q = 0, 1, 2$	forme rationnelle	4 additions 2 multipl. 2 divisions	17	94	200	$2 \cdot 10^{-9}$	

REMARQUES :

- Pour tous les sous-programmes, l'intervalle de variation de l'argument est $(-10^{10}, 10^{10})$
- Dans SFATT2 et SFATR2, nous appliquons la formule $\operatorname{Arctg} x = \frac{\pi}{2} - \operatorname{Arctg} \frac{1}{x}$ quand x est supérieur à 1. Ceci permet de réduire le nombre de constantes précalculées (voir paragraphe 122-411-1).

D'après ce tableau comparatif, nous voyons qu'aucune méthode ne l'emporte nettement sur les autres. Nous retenons néanmoins SFATS qui donne le meilleur temps.

Perte de précision

Référence	Erreur de méthode	η
SFATT1	$2,3 \cdot 10^{-10}$	$1 \cdot 10^{-9}$
SFATT2	$2,3 \cdot 10^{-10}$	$2 \cdot 10^{-9}$
SFATS	$1,2 \cdot 10^{-10}$	$2 \cdot 10^{-9}$
SFATH	$4 \cdot 10^{-8}$	$4 \cdot 10^{-8}$
SFATR1	$2,5 \cdot 10^{-11}$	$1 \cdot 10^{-9}$
SFATR2	$2,5 \cdot 10^{-11}$	$2 \cdot 10^{-9}$

L'étude de la précision porte sur le programme retenu SFATS. Nous nous plaçons dans le cas où l'argument est supérieur à 1. En effet, il s'agit d'un cas défavorable puisque nous commençons les calculs avec l'inverse de l'argument, c'est-à-dire, avec une donnée entachée d'erreur.

Posons $X = \frac{1}{x}$; l'erreur sur X correspond à l'erreur de chute après la division, soit $\Delta(X) < 1 \cdot 10^{-10}$ (X peut être cadré 0). Le changement de variable $t = \frac{X - X_0}{1 + XX_0}$ permet le cadrage

$0(t < 1)$; il en résulte une erreur de chute inférieure à $1 \cdot 10^{-10}$ à laquelle il faut ajouter l'erreur provenant de l'imprécision de X :

$$\frac{\Delta(X - X_0)}{1 + XX_0} + \frac{(X - X_0)}{(1 + XX_0)^2} \Delta(1 + XX_0)$$

$$\text{avec } |X - X_0| \leq 0,05, \quad |X_0| \leq 0,95 \quad \Delta(X - X_0) \leq 10^{-10}$$

L'erreur sur $1 + XX_0$ est la somme de $X_0 \Delta(X)$ et de l'erreur de chute de $1 + XX_0$ cadré 1.

Nous obtenons $\Delta(1 + XX_0) \leq 5,95 \cdot 10^{-10}$; l'erreur totale sur t est alors inférieure à $2,3 \cdot 10^{-10}$.

En tenant compte de la petitesse de t ($|t| \leq 0,05$), l'erreur de calcul sur l'approximation est de l'ordre de celle sur t . Mais l'approximation est obligatoirement cadrée 1, il en résulte une erreur de chute de $1 \cdot 10^{-9}$ soit $\Delta(A) \leq 12,3 \cdot 10^{-10}$.

L'erreur inhérente à $\text{Arctg } X_0$ (cadré 1) est inférieure à $4 \cdot 10^{-10}$, celle sur $\frac{\pi}{2}$ à $2 \cdot 10^{-10}$.

L'erreur globale a donc pour limite supérieure $1,83 \cdot 10^{-9}$.

REMARQUE : L'erreur sur les calculs ne dépasse pas $1,5 \cdot 10^{-9}$ (au lieu de $1,83 \cdot 10^{-9}$) quand l'argument est inférieur à l'unité.

225 - 2 Double précision

TABLEAU DE COMPARAISON

Référence	Intervalle réduit	Approximation		C ^{tes}	E	K	η	Référence dans la bibliothèque
DFATS	$(-1, 1)$	6 termes	7 multipl.	17	195	1100	$5 \cdot 10^{-19}$	DFAT
DFAT.T	$(-tg \frac{\pi}{16}, tg \frac{\pi}{16})$	9 termes	10 multipl.	non programmé				
DFAT.R	$(-tg \frac{\pi}{16}, tg \frac{\pi}{16})$	forme rationnelle	10 multipl. 1 division	non programmé				

Nous n'avons programmé que le développement en série avec relaxation. En effet, la réduction d'intervalle est plus simple et demande moins d'instructions ; le développement converge plus rapidement comme le montre le tableau.

Perte de précision

En fin de programme, avant le renvoi à l'ADP, le résultat est en virgule fixe, cadré 1 à compter de la position 20 de l'accumulateur. L'erreur relevée est inférieure à $5 \cdot 10^{-19}$.

Référence	Erreur de méthode	η
DFATS	$2,3 \cdot 10^{-21}$	$5 \cdot 10^{-19}$

L'étude de la précision, exposée au paragraphe 225 - 1, a été reprise sur DFATS. En tenant compte des erreurs de chute et des erreurs dues aux données, nous avons trouvé une erreur de l'ordre de $3,6 \cdot 10^{-19}$. Nous voyons que l'erreur de méthode est négligeable devant cette erreur.

226 - Arc Sinus (Simple précision)

TABLEAU DE COMPARAISON

Référence	Intervalle de variation	Changement de variable	Approximation		Ctes	E	K	η	Référence dans la bibliothèque
SFAST	$(0, \frac{\sqrt{2}}{2})$	$X = x \sqrt{2}$	10 termes	9 addit. 11 mult.	16	99	235	$5 \cdot 10^{-10}$	SFAS
	$(\frac{\sqrt{2}}{2}, 1)$	$X = \sqrt{2(1-x^2)}$	11 termes	10 addit. 11 mult.			460	$3 \cdot 10^{-9}$	
SFASH	$(0, 1)$	$X = \sqrt{1-x}$	9 termes	8 addit. 8 multipl.	11	72	380	$2 \cdot 10^{-8}$	

REMARQUE : Pour les deux sous-programmes, l'intervalle de variation de l'argument est $(-1, 1)$.

Dès qu'il faut introduire une racine, le temps augmente de 62 à 96 % suivant les approximations. Pour cette raison, nous retenons SFAST puisque, sur la plus grande partie de l'intervalle de variation, à savoir $(0 ; 0,7)$, il donne les meilleurs temps et précision.

Précision

Nous avons vérifié expérimentalement que, pour des valeurs voisines de l'unité, le développement d'Hastings donne une précision très supérieure à celle qui est obtenue par le développement de Tchebycheff.

Nous donnons quelques valeurs d'essai dans le tableau suivant :

Valeur de l'argument	Erreur sur le résultat	Erreur sur le résultat
	Tchebycheff	Hastings
$\sin 60^\circ = 0,866025404$	$3 \cdot 10^{-9}$	$8 \cdot 10^{-9}$
$\sin 70^\circ = 0,939692621$	$3 \cdot 10^{-9}$	$2 \cdot 10^{-9}$
$\sin 80^\circ = 0,984807753$	$3,5 \cdot 10^{-9}$	$5 \cdot 10^{-10}$
$\sin 89^\circ 30' = 0,999961923$	$2,5 \cdot 10^{-8}$	$5 \cdot 10^{-10}$

En effet, pour x supérieur à $\frac{\sqrt{2}}{2}$, nous effectuons le changement de variable $X = \sqrt{2(1-x^2)}$ dans le développement de Tchebycheff qui se présente alors sous la forme :

$$\frac{\pi}{2} - \sum_{n=0}^8 K_{2n+1} X^{2n+1}$$

L'erreur sur ce développement est :

$$\Delta(T) = \Delta\left(\frac{\pi}{2}\right) + \sum_{n=0}^8 X^{2n+1} \Delta(K_{2n+1}) + \Delta(X) \sum_{n=0}^8 (2n+1) K_{2n+1} X^{2n}$$

avec $\Delta\left(\frac{\pi}{2}\right) = 2 \cdot 10^{-10}$, $\Delta(K_{2n+1}) = 5 \cdot 10^{-11}$ (Les K_{2n+1} sont cadrés 0).

Posons $x = 1 - \alpha$. Pour α très petit, X est sensiblement égal à $2\sqrt{\alpha}$, c'est-à-dire, très voisin de 0.

Par suite, le terme $\sum_{n=0}^8 X^{2n+1} \Delta(K_{2n+1})$ est négligeable devant les autres.

Le terme $\sum_{n=0}^8 (2n+1) K_{2n+1} X^{2n}$ se réduit sensiblement à $K_1 \simeq \frac{\sqrt{2}}{2}$

$\Delta(X)$ qui est égal à $\sqrt{2} \Delta(\sqrt{1-x^2}) + \sqrt{1-x^2} \Delta(\sqrt{2})$ est de l'ordre de $\sqrt{2} \Delta(\sqrt{1-x^2})$

d'où : $\Delta(T) \simeq 2.10^{-10} + \Delta(\sqrt{1-x^2})$ quand x est voisin de 1.

Quant au développement d'Hastings, il se présente sous la forme : $\frac{\pi}{2} - \sqrt{1-x} \sum_{n=0}^7 K_n x^n$

L'erreur sur ce développement est :

$$\Delta(H) = \Delta\left(\frac{\pi}{2}\right) + \sqrt{1-x} \left[\sum_{n=0}^7 x^n \Delta(K_n) + \Delta(x) \sum_{n=1}^7 n K_n x^{n-1} \right] + \Delta(\sqrt{1-x}) \sum_{n=0}^7 K_n x^n$$

$$\text{avec } \Delta\left(\frac{\pi}{2}\right) = 2.10^{-10} ; \Delta(x) = 0 ; \Delta(K_n) = 5.10^{-11}$$

Si α est très petit, le terme $5.10^{-11} \cdot \sqrt{1-x} \sum_{n=0}^7 x^n$ est négligeable devant les autres.

Le terme $\sum_{n=0}^7 K_n x^n$ tend vers $\sqrt{2}$ d'où :

$$\Delta(H) \simeq 2.10^{-10} + \sqrt{2} \Delta(\sqrt{1-x}) \text{ quand } x \text{ est voisin de 1.}$$

Compte tenu du matériel utilisé et du cadrage 11 de l'argument à l'entrée, x a pour limite supérieure :

$$0,99999999 \text{ soit } \alpha = 10^{-9}$$

(Un test permet d'éliminer le cas où $x = 1$)

A la limite $\alpha = 10^{-9}$, pour le développement d'Hastings, l'erreur est :

$$\Delta(H) = 2.10^{-10} + \sqrt{2} \frac{\Delta(1-x)}{2\sqrt{10^{-9}}}$$

mais le terme $1-x$ est connu exactement et par suite $\Delta(H) = 2.10^{-10}$

Pour le développement de Tchébycheff, l'erreur devient : $\Delta(T) = 2.10^{-10} + \frac{\Delta(1-x^2)}{2\sqrt{2.10^{-9}}}$

La valeur $1-x^2$ est systématiquement entachée d'erreur. $1-x^2$ est cadré 0 mais avec $\alpha = 10^{-9}$, il y a obligatoirement 18 décimales. La troncature à 10 décimales, la dernière étant arrondie, entraîne une erreur inférieure ou égale à 5.10^{-11}

$$\Delta(T) = 2.10^{-10} + 5,6 \cdot 10^{-7}$$

Aussi, nous voyons que, pour les valeurs de l'argument très proches de l'unité, il est préférable d'utiliser le développement d'Hastings. Le passage par l'Arc tangente donnerait sensiblement la même erreur que le développement de Tchebycheff.

23 - NOTICES D'UTILISATION DES SOUS-PROGRAMMES DES BIBLIOTHEQUES

231 - Sous-programmes fonctionnels en simple précision

Un sous-programme fonctionnel « simple précision » est utilisable indifféremment en virgule fixe ou flottante et ne requiert aucun autre sous-programme auxiliaire.

Une première partie des cartes se présente sous la forme numérique translatable « 1 mot par carte ». Le facteur de translation Δ , qui peut être quelconque mais limité, est chargé en 1950. Ces cartes contiennent les instructions qui permettent le calcul de la fonction en virgule fixe ou celui de sa mantisse en virgule flottante.

L'autre partie, placée directement à la suite, est « chargement 1 mot » et concerne les instructions supplémentaires nécessaires à la virgule flottante. Ces instructions sont enregistrées obligatoirement dans la zone (1735 - 1799) qui précède immédiatement celle qu'occupe le sous-programme A V F (Arithmétique en Virgule Flottante simple précision)

ceci présente trois avantages :

- a) faciliter la manipulation : nous utilisons le même sous-programme aussi bien en virgule fixe que flottante,
- b) libérer la zone (1735 - 1799) dans le cas de la virgule fixe. Il faudra alors enregistrer le sous-programme fonctionnel avant tout autre programme utilisant cette zone,
- c) réduire l'encombrement quand nous avons plusieurs fonctions à calculer en virgule flottante : La zone (1735 - 1799) est commune à toutes les fonctions.

231 - 1 Sous-programmes « Racine Carrée »

Il existe deux sous-programmes « Racine Carrée » référés SFRCE et SFRCK, le premier est condensé, le second rapide.

231 - 11 Sous-programme SFRCE

231 - 111 Utilisation en virgule fixe

OBJET : Ce sous-programme calcule la racine carrée en virgule fixe et simple précision avec un encombrement restreint.

CONDITIONS D'UTILISATION :

- Les limites de l'argument sont : $0 \leq \text{Argument} < 10^{10}$
- L'argument ne doit pas avoir plus de 10 chiffres significatifs. S'il en a davantage, le sous-programme n'offre aucune sécurité.

Utilisation

Entrée :

- L'argument est cadré 11 dans les accumulateurs, à partir de l'extrême gauche.
- Le distributeur contient une instruction de sortie.

La première instruction du sous-programme a pour adresse : $0000 + \Delta$

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1685 ; il est chargé en 1950.

Sortie :

- 8003 : 00 0000 0000
- 8002 : Y,Y YYYY YYYY +
- 8001 : 00 0000 000 $n_{\pm}$

La racine carrée, ramenée dans l'intervalle $(1, 10)$ se trouve en 8002, cadrée 1 (La virgule a même position qu'à l'entrée).

La position « unité » de 8001 indique la puissance de 10 (avec son signe) qui doit multiplier le résultat contenu en 8002.

Le résultat final s'écrit : Y,Y YYYY YYYY $\cdot 10^{\pm n}$

REMARQUE : Cette présentation du résultat permet de conserver 10 chiffres significatifs dans le cas où l'argument est inférieur à l'unité.

	8003	8002	8001
Entrée	00 0000 0000	0,0 0020 0000 +	1.5
Sortie	00 0000 0000	1,4 1421 3562 +	00 0000 0002 ⁻
Soit $\sqrt{0,0002} = 1,414213562 \cdot 10^{-2}$			

CARACTERISTIQUES :

- Encombrement : 49 mémoires de Δ à $\Delta + 0048$
- Temps de calcul: 365 millisecondes environ
- Précision : l'erreur absolue est inférieure à $1 \cdot 10^{-9}$ sur le contenu de 8002
- Arrêt programmé: 01 0011 9999 si l'argument est négatif
- Méthode utilisée: méthode itérative de Newton (voir paragraphe 111 - 1)

231 - 112 Utilisation en virgule flottante

OBJET : Ce sous-programme calcule la racine carrée en virgule flottante et simple précision, avec un encombrement restreint.

CONDITIONS D'UTILISATION :

- L'argument, positif, est écrit en virgule flottante du type FLAIR (exposant à gauche)

L'unité est alors représentée par : 51 1000 0000 +

- L'exposant de l'argument doit avoir pour limites : $41 < \alpha\alpha < 61$

REMARQUE 1 : Cette limitation de l'exposant n'est pas, en fait, restrictive pour la racine carrée. Il suffit d'additionner $2k$ (k positif ou négatif) à l'exposant de l'argument pour le ramener dans les limites imposées, l'exposant du résultat est alors $\beta\beta + k$; la mantisse reste inchangée.

REMARQUE 2 : Pour $\alpha\alpha \geq 61$, il y a arrêt programmé car le résultat obtenu serait nécessairement erroné par suite de la perte du ou des premiers chiffres de la mantisse.

Pour $\alpha\alpha = 41$, l'argument est arrondi à 0,0 0000 0001 si le premier chiffre de la mantisse est supérieur ou égal à 5. Dans le cas contraire, l'argument est considéré comme nul.

Pour $\alpha\alpha < 41$, l'argument est considéré comme nul.

Utilisation

Entrée :

- La mémoire d'adresse 1778 doit contenir la constante d'orientation : 69 1757 Δ +

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1685. Il est chargé en 1950.

Le contenu de 1778 n'est pas modifié par l'exécution du sous-programme.

- 8003 : $\alpha\alpha$ XXXX XXXX + (Argument)

- 8002 : 00 0000 0000

- 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 1750

Sortie :

- 8003 : $\beta\beta$ YYYY YYYY + (Racine carrée)

- 8002 : 00 0000 0000

- 8001 : $\beta\beta$ 0000 0000 + si le résultat n'est pas nul
quelconque si le résultat est nul

CARACTERISTIQUES :

- Encombrement : 114 mémoires de Δ à $\Delta + 0048$ et de 1735 à 1799

- Temps de calcul : 475 millisecondes environ

- Précision : l'erreur sur la mantisse du résultat est inférieure ou égale à $5,10^{-9}$ avec la convention :
 $0,1 \leq \text{mantisse} < 1$

- Arrêts programmés : 01 1745 9999 si l'argument a un exposant supérieur ou égal à 61
01 0011 9999 si l'argument est négatif

- Méthode utilisée : méthode itérative de Newton (voir paragraphe 111 - 1)

231 - 12 Sous-programme SFRCK

231 - 121 Utilisation en virgule fixe

OBJET : Ce sous-programme calcule la racine carrée en virgule fixe et simple précision, avec un temps de calcul réduit.

CONDITIONS D'UTILISATION :

- Les limites de l'argument sont : $0 \leq \text{Argument} < 10^{10}$
- L'argument ne doit pas avoir plus de 10 chiffres significatifs. S'il en a davantage, le sous-programme n'offre aucune sécurité.

Utilisation

Entrée :

- L'argument est cadré 11 dans les accumulateurs à partir de l'extrême gauche
- Le distributeur contient une instruction de sortie
La première instruction du sous-programme a pour adresse : $0000 + \Delta$
Le facteur de translation Δ peut être quelconque, pourvu qu'il reste inférieur ou égal à 1630 ; il est chargé en 1950.

Sortie :

- 8003 : 00 0000 0000
- 8002 : Y,Y YYYY YYYY +
- 8001 : 00 0000 000n $\pm$.

La racine carrée, ramenée dans l'intervalle $(1, 10)$ se trouve en 8002, cadrée 1 (La virgule a même position qu'à l'entrée).

La position « unité » de 8001 indique la puissance de 10 (avec son signe) qui doit multiplier le résultat contenu en 8002.

Le résultat final s'écrit : Y,Y YYYY YYYY . $10^{\pm n}$

CARACTERISTIQUES :

- Encombrement : 105 mémoires de Δ à $\Delta + 0104$
- Temps de calcul : 160 millisecondes
- Précision : l'erreur absolue est inférieure ou égale à $2 \cdot 10^{-9}$ sur le contenu de 8002
- Arrêt programmé : 01 0011 9999 si l'argument est négatif
- Méthode utilisée : la racine est approchée par une expression rationnelle (voir paragraphe 111 - 2)

231 - 122 Utilisation en virgule flottante

OBJET : Ce sous-programme calcule la racine carrée en virgule flottante et simple précision, avec un temps de calcul réduit.

CONDITIONS D'UTILISATION :

- L'argument, positif, est écrit en virgule flottante du type FLAIR (exposant à gauche)
- L'unité est alors représentée par : 51 1000 0000 +
- L'exposant de l'argument doit avoir pour limites : $41 < \alpha < 61$

(Voir les remarques du paragraphe 231 - 112)

Utilisation

Entrée :

- La mémoire d'adresse 1778 doit contenir la constante d'orientation : 69 1757 Δ +

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1630. Il est chargé en 1950.

Le contenu de 1778 n'est pas modifié par l'exécution du sous-programme.

- 8003 : $\alpha\alpha$ XXXX XXXX + (Argument)
- 8002 : 00 0000 0000
- 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 1750

Sortie :

- 8003 : $\beta\beta$ YYYY YYYY + (Racine carrée)
- 8002 : 00 0000 0000
- 8001 : $\beta\beta$ 0000 0000 + si le résultat n'est pas nul
quelconque si le résultat est nul

CARACTERISTIQUES :

- Encombrement : 170 mémoires de Δ à Δ + 0104 et de 1735 à 1799
- Temps de calcul : 270 millisecondes
- Précision : l'erreur sur la mantisse du résultat est inférieure ou égale à $5 \cdot 10^{-9}$ avec la convention $0,1 \leq \text{mantisse} < 1$
- Arrêts programmés: 01 1745 9999 si l'argument a un exposant supérieur ou égal à 61
01 0011 9999 si l'argument est négatif
- Méthode utilisée : la racine est approchée par une expression rationnelle (voir paragraphe 111 - 2)

231 - 2 Sous-programme « Logarithme Népérien » (Référence SFLE)

231 - 21 Utilisation en virgule fixe

OBJET : Ce sous-programme calcule le logarithme népérien en virgule fixe et simple précision.

CONDITIONS D'UTILISATION :

- Les limites de l'argument sont : $10^{-9} \leq \text{Argument} < 10^{10}$
- L'argument ne doit pas avoir plus de 10 chiffres significatifs. S'il en a davantage, le sous-programme n'offre aucune sécurité.

Utilisation

Entrée :

- L'argument est cadré 11 dans les accumulateurs à partir de l'extrême gauche
- Le distributeur contient une instruction de sortie.

La première instruction du sous-programme a pour adresse : $0000 + \Delta$

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1640 ; il est chargé en 1950.

Sortie :

- 8000 : 00 0000 000Y Y,Y YYYY YYYY ±
- 8001 : quelconque

Le logarithme népérien est cadré 11 dans les accumulateurs. Compte tenu des limites de l'argument, le résultat peut avoir jusqu'à 11 chiffres significatifs.

CARACTERISTIQUES :

- Encombrement : 91 mémoires de Δ à $\Delta + 0090$
- Temps de calcul : 250 millisecondes
- Précision : l'erreur absolue est inférieure ou égale à $2 \cdot 10^{-9}$ sur le résultat
- Arrêt programmé : 01 0012 9999 si l'argument est négatif ou nul
- Méthode utilisée : développement en série limité et relaxation (voir paragraphe 122 - 111 - 2)

231 - 22 Utilisation en virgule flottante

OBJET : Ce sous-programme calcule le logarithme népérien en virgule flottante et simple précision.

CONDITIONS D'UTILISATION :

- L'argument est écrit en virgule flottante du type FLAIR (exposant à gauche)
- L'unité est représentée par : 51 1000 0000 +

- L'argument doit être positif et les limites de son exposant sont : $41 < \alpha\alpha < 61$

REMARQUE : Si le premier chiffre de la mantisse est supérieur ou égal à 5 et si $\alpha\alpha = 41$, l'argument est arrondi à 42 1000 0000 +

Utilisation

Entrée :

- La mémoire d'adresse 1778 doit contenir la constante d'orientation : 69 1757 Δ —

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1640 ; il est chargé en 1950 .

Le contenu de 1778 n'est pas modifié après l'exécution du sous-programme

- 8003 : $\alpha\alpha$ XXXX XXXX + (Argument)

- 8002 : 00 0000 0000

- 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 1750

Sortie :

- 8003 : $\beta\beta$ YYYY YYYY $\pm$ (Logarithme)

- 8002 : 00 0000 0000

- 8002 : $\beta\beta$ 0000 0000 + si le résultat n'est pas nul
quelconque si le résultat est nul

CARACTERISTIQUES

- Encombrement : 156 mémoires de Δ à $\Delta + 0090$ et de 1735 à 1799

- Temps de calcul : 360 millisecondes

- Précision : l'erreur sur la mantisse du résultat est inférieure ou égale à $2 \cdot 10^{(41-\beta\beta)}$ avec la convention $0,1 \leq \text{mantisse} < 1$. Pour $\beta\beta$ supérieur à 49, la mantisse est exacte .

- Arrêts programmés: 01 1745 9999 si l'exposant de l'argument est supérieur ou égal à 61

01 0012 9999 si l'argument est négatif ou nul ou si son exposant est inférieur ou égal à 41

- Méthode utilisée : développement en série limité et relaxation (voir première partie, paragraphe 122-111-2)

231 - 3 Sous-programmes « Exponentielle »

Il existe deux sous-programmes « Exponentielle » référés SFEX E et SFEX K :

le premier est condensé, le second rapide .

231 - 31 Sous-programme SFEXE

231 - 311 Utilisation en virgule fixe

OBJET : Ce sous-programme calcule l'exponentielle en virgule fixe et simple précision, avec un encombrement restreint.

CONDITIONS D'UTILISATION :

- Les limites de l'argument sont : $|\text{Argument}| < 10^{10}$
- L'argument peut avoir jusqu'à 19 chiffres significatifs pourvu que le nombre de chiffres significatifs avant la virgule ne dépasse pas 10.

Utilisation

Entrée :

- L'argument est cadré 11 dans les accumulateurs à partir de l'extrême gauche

8000 : 0X XXXX XXXX X,X XXXX XXXX ±

- Le distributeur contient une instruction de sortie.

La première instruction du sous-programme a pour adresse : 0000 + Δ

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1670 ; il est chargé en 1950

Sortie :

- 8003 : 00 0000 0000
- 8002 : Y,Y YYYY YYYY +
- 8001 : nn nnnn nnnn ±

L'exponentielle, ramenée dans l'intervalle (1,10) se trouve en 8002, cadrée 1

8001 contient la valeur et le signe de la puissance de 10 par laquelle le contenu de 8002 doit être multiplié. Le chiffre des unités se trouve en position 1. Les 10 positions peuvent être occupées.

Le résultat final s'écrit : Y,Y YYYY YYYY . $10^{\pm N}$

CARACTERISTIQUES

- Encombrement : 65 mémoires de Δ à $\Delta + 0064$
- Temps de calcul : 330 millisecondes
- Précision : l'erreur absolue est inférieure ou égale à $3 \cdot 10^{-9}$ sur le contenu de 8002
- Méthode utilisée : approximation rationnelle par la série de Darboux (voir paragraphe 122 - 212)

231 - 312 Utilisation en virgule flottante

OBJET : Ce sous-programme calcule l'exponentielle en virgule flottante et simple précision, avec un encombrement restreint.

CONDITIONS D'UTILISATION

- L'argument est écrit en virgule flottante du type FLAIR (exposant à gauche)

L'unité est représentée par : 51 1000 0000 +

- Les limites de l'argument sont en écriture « virgule flottante » :

$$| \text{Argument} | \leq 53 \ 1120 \ 0000$$

REMARQUE : Un argument en dehors de ces limites conduit à un résultat en virgule flottante pour lequel l'exposant est, soit négatif, soit supérieur à 99.

Utilisation

Entrée :

- La mémoire d'adresse 1778 doit contenir la constante d'orientation : 69 1757 Δ +
Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1670 ; il est chargé en 1950. Le contenu de 1778 n'est pas modifié par l'exécution du sous-programme.
- 8003 : $\alpha\alpha \ XXXX \ XXXX \pm (\text{Argument})$
- 8002 : 00 0000 0000
- 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 1750

Sortie :

- 8003 : $\beta\beta \ YYYYY \ YYYYY \ + (\text{Exponentielle})$
- 8002 : 00 0000 0000
- 8001 : $\beta\beta \ 0000 \ 0000 \ +$

CARACTERISTIQUES

- Encombrement : 130 mémoires de Δ à 0064 + Δ et de 1735 à 1799
- Temps de calcul : 440 millisecondes
- Précision : l'erreur sur la mantisse du résultat est inférieure ou égale à $5 \cdot 10^{-9}$ avec la convention : $0,1 \leq \text{mantisse} < 1$ (l'erreur est indépendante de l'exposant $\beta\beta$)
- Arrêt programmé : 01 1745 9999 si $| \text{Argument} | > 53 \ 1120 \ 0000$
- Méthode utilisée : approximation rationnelle par la série de Darboux (voir première partie, par. 122-212)

231 - 32 Sous programme SFEX K

231 - 321 Utilisation en virgule fixe

OBJET : Ce sous-programme calcule l'exponentielle en virgule fixe et simple précision, avec un temps de calcul réduit.

CONDITIONS D'UTILISATION

- Les limites de l'argument sont : $| \text{Argument} | < 10^{10}$

- L'argument peut avoir jusqu'à 19 chiffres significatifs pourvu que le nombre de chiffres significatifs avant la virgule ne dépasse pas 10 .

Utilisation

Entrée :

- L'argument est cadré 11 dans les accumulateurs à partir de l'extrême gauche

8000 : 0X XXXX XXXX X,X XXXX XXXX ±

- Le distributeur contient une instruction de sortie

La première instruction du sous-programme a pour adresse : 0000 + Δ

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1650 ; il est chargé en 1950

Sortie :

- 8003 : 00 0000 0000

- 8002 : Y,Y YYYY YYYY +

- 8001 : nn nnnn nnnn ±

L'exponentielle, ramenée dans l'intervalle (1,10), se trouve en 8002, cadrée 1
8001 contient la valeur et le signe de la puissance de 10 par laquelle le contenu de 8002 doit être multiplié. Le chiffre des unités se trouve en position 1. Les 10 positions peuvent être occupées .

Le résultat final s'écrit : Y,Y YYYY YYYY . $10^{\pm N}$

CARACTERISTIQUES :

- Encombrement : 81 mémoires de Δ à $\Delta + 0080$
- Temps de calcul : 200 millisecondes
- Précision : l'erreur absolue est inférieure ou égale à $6 \cdot 10^{-9}$ sur le contenu de 8002
- Méthode utilisée : développement en série et relaxation (voir paragraphe 122-211-2)

231 - 322 Utilisation en virgule flottante

OBJET : Ce sous-programme calcule l'exponentielle en virgule flottante et simple précision, avec un temps de calcul réduit.

CONDITIONS D'UTILISATION :

- L'argument est écrit en virgule flottante du type FLAIR (exposant à gauche)
L'unité est représentée par : 51 1000 0000 +
- Les limites de l'argument sont, en écriture « virgule flottante » : | Argument | ≤ 53 1120 0000

REMARQUE : Un argument en dehors de ces limites conduit à un résultat en virgule flottante pour lequel l'exposant est négatif ou supérieur à 99 .

Utilisation

Entrée :

- La mémoire d'adresse 1778 doit contenir la constante d'orientation : $69\ 1757\ \Delta +$
Le facteur de translation Δ peut être quelconque, pourvu qu'il reste inférieur ou égal à 1650 ; il est chargé en 1950
Le contenu de 1778 n'est pas modifié par l'exécution du sous-programme

- 8003 : $\alpha\alpha\ XXXX\ XXXX \pm (\text{Argument})$

- 8002 : 00 0000 0000

- 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 1750

Sortie :

- 8003 : $\beta\beta\ YYYY\ YYYY + (\text{Exponentielle})$

- 8002 : 00 0000 0000

- 8001 : $\beta\beta\ 0000\ 0000 +$

CARACTERISTIQUES

- Encombrement : 146 mémoires de Δ à $\Delta + 0080$ et de 1735 à 1799
- Temps de calcul : 310 millisecondes
- Précision : l'erreur sur la mantisse du résultat est inférieure ou égale à $5 \cdot 10^{-9}$ avec la convention : $0,1 \leq \text{mantisse} < 1$ (l'erreur est indépendante de l'exposant $\beta\beta$)
- Arrêt programmé : 01 1745 9999 si $|\text{Argument}| > 53\ 1120\ 0000$
- Méthode utilisée : développement en série limité et relaxation (voir première partie, par. 122-211-2)

231 - 4 Sous-programme « Sinus et Cosinus » (référence SFS C)

231 - 41 Utilisation en virgule fixe

OBJET : Ce sous-programme calcule le sinus ou le cosinus en virgule fixe et simple précision.

CONDITIONS D'UTILISATION :

- Les limites de l'argument sont : $|\text{Argument}| < 10^{10}$
- L'argument peut avoir jusqu'à 19 chiffres significatifs à condition toutefois que le nombre de chiffres significatifs avant la virgule ne dépasse pas 10.

UtilisationEntrée :

- L'argument est cadré 11 dans les accumulateurs à partir de l'extrême gauche

8000 : 0X XXXX XXXX X,X XXXX XXXX±

- Le distributeur contient une instruction de sortie

La première instruction du sous-programme a pour adresse :

0000 + Δ pour le calcul du sinus

0001 + Δ pour le calcul du cosinus

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1680 ; il est chargé en 1950 .

Sortie :

- 8003 : 00 0000 0000
- 8002 : Y,Y YYYY YYYY± (Sinus ou Cosinus)
- 8001 : quelconque

CARACTERISTIQUES :

- Encombrement : 53 mémoires de Δ à $\Delta + 0052$
- Temps de calcul : 155 millisecondes
- Précision : l'erreur absolue sur le résultat est inférieure ou égale à $2 \cdot 10^{-9}$
- Méthode utilisée : développement limité de Tchebycheff de $\cos \frac{\pi}{2} X$ (voir paragraphe 122-311-12)

231 - 42 Utilisation en virgule flottante

OBJET : Ce sous-programme calcule le sinus ou le cosinus en virgule flottante et simple précision.

CONDITIONS D'UTILISATION :

- L'argument est écrit en virgule flottante du type FLAIR (exposant à gauche)
L'unité est représentée par : 51 1000 0000 +
- L'exposant de l'argument doit être inférieur à 61

UtilisationEntrée :

- La mémoire d'adresse 1778 doit contenir la constante d'orientation :

69 1757 Δ ← pour le calcul du sinus

69 1757 $\Delta + 0001$ ← pour le calcul du cosinus

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1680 ; il est chargé en 1950

Le contenu de 1778 n'est pas modifié par l'exécution du sous-programme

- 8003 : $\alpha\alpha$ XXXX XXXX $\pm$ (Argument)
- 8002 : 00 0000 0000
- 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 1750

Sortie :

- 8003 : $\beta\beta$ YYYY YYYY $\pm$ (Sinus ou Cosinus)
- 8002 : 00 0000 0000
- 8001 : $\beta\beta$ 0000 0000 + si le résultat n'est pas nul
: quelconque si le résultat est nul

CARACTERISTIQUES :

- Encombrement : 118 mémoires de Δ à $\Delta + 0052$ et de 1735 à 1799
- Temps de calcul : 265 millisecondes
- Précision : l'erreur sur la mantisse du résultat est inférieure ou égale à $2 \cdot 10^{41-\beta\beta}$ avec la convention : $0,1 \leq \text{mantisse} < 1$. Pour $\beta\beta = 50$ ou 51, la mantisse est exacte
- Arrêt programmé : 01 1745 9999 si l'exposant de l'argument est supérieur ou égal à 61
- Méthode utilisée : développement limité de Tchebycheff de $\cos \frac{\pi}{2} X$ (voir première partie, paragraphe 122 - 311 - 12)

231 - 5 Sous-programme « Arc tangente » (référence : SFAT)

231 - 51 Utilisation en virgule fixe

OBJET : Ce sous-programme calcule l'Arc tangente en virgule fixe et simple précision .

CONDITIONS D'UTILISATION :

- Les limites de l'argument sont : $|\text{Argument}| < 10^{10}$
- L'argument ne doit pas avoir plus de 10 chiffres significatifs. S'il en a davantage, le sous-programme n'offre aucune sécurité.

Utilisation

Entrée :

- L'argument est cadré 11 dans les accumulateurs à partir de l'extrême gauche .

- Le distributeur contient une instruction de sortie
La première instruction du sous-programme a pour adresse : 0000 + Δ
Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1645 ; il est chargé en 1950

Sortie :

- 8003 : 00 0000 0000
- 8002 : Y,Y YYYY YYYY $\pm$ (Arc tangente)
- 8001 : quelconque

CARACTERISTIQUES :

- Encombrement : 86 mémoires de Δ à $\Delta + 0085$
- Temps de calcul : 175 millisecondes
- Précision : l'erreur absolue sur le résultat est inférieure ou égale à $2 \cdot 10^{-8}$
- Méthode utilisée : développement en série et relaxation (voir paragraphe 122 - 411 - 2)

231 - 52 Utilisation en virgule flottante

OBJET : Ce sous-programme calcule l'Arc tangente en virgule flottante et simple précision

CONDITIONS D'UTILISATION :

- L'argument est écrit en virgule flottante du type FLAIR (exposant à gauche)
L'unité est alors représentée par : 51 1000 0000 +
- L'exposant de l'argument doit être inférieur à 61

Utilisation

Entrée :

- La mémoire d'adresse 1778 doit contenir la constante d'orientation : 69 1757 Δ —
Le facteur de translation peut être quelconque pourvu qu'il reste inférieur ou égal à 1645 ; il est chargé en 1950
Le contenu de 1778 n'est pas modifié par l'exécution du sous-programme
- 8003 : $\alpha\alpha$ XXXX XXXX $\pm$ (Argument)
- 8002 : 00 0000 0000
- 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 1750

Sortie :

- 8003 : $\beta\beta$ YYY YYY $\pm$ (Arc tangente)
- 8002 : 00 0000 0000
- 8001 : $\beta\beta$ 0000 0000 + si le résultat n'est pas nul
: quelconque si le résultat est nul

CARACTERISTIQUES :

- Encombrement : 151 mémoires de Δ à $\Delta + 0085$ et de 1735 à 1799
- Temps de calcul : 285 millisecondes
- Précision : l'erreur sur la mantisse du résultat est inférieure ou égale à $2 \cdot 10^{41-\beta\beta}$ avec la convention : $0,1 \leq \text{mantisse} < 1$. Pour $\beta\beta = 50$ ou 51, la mantisse est exacte.
- Arrêt programmé : 01 1745 9999 si l'exposant de l'argument est supérieur ou égal à 61
- Méthode utilisée : développement en série et relaxation (voir paragraphe 122 - 411 - 2)

231 - 6 Sous-programme « Arc sinus » (référence : SFAS)

231 - 61 Utilisation en virgule fixe

OBJET : Ce sous-programme calcule l'Arc sinus en virgule fixe et simple précision

CONDITIONS D'UTILISATION :

- La valeur absolue de l'argument doit être inférieure ou égale à l'unité : $|\text{Argument}| \leq 1$

Utilisation

Entrée :

- 8003 : 00 0000 0000
- 8002 : X,X XXXX XXXX $\pm$ (Argument)
- 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 0000 + Δ

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1635 ;
il est chargé en 1950

Sortie :

- 8003 : 00 0000 0000
- 8002 : Y,Y YYY YYY $\pm$ (Arc sinus)
- 8001 : quelconque

CARACTERISTIQUES :

- Encombrement : 99 mémoires de Δ à $\Delta + 0098$
- Temps de calcul : 235 millisecondes si la valeur absolue de l'argument est inférieure ou égale à $\frac{\sqrt{2}}{2}$,
460 millisecondes si la valeur absolue de l'argument est supérieure à $\frac{\sqrt{2}}{2}$
- Précision : - si la valeur absolue du résultat est inférieure ou égale à $\frac{\pi}{4}$, l'erreur est de $5 \cdot 10^{-10}$
- si la valeur absolue du résultat est supérieure à $\frac{\pi}{4}$, l'erreur est de $3 \cdot 10^{-9}$
sauf au voisinage de $\frac{\pi}{2}$ où elle est de $5 \cdot 10^{-7}$
- Arrêt programmé : 01 0015 9999 si la valeur absolue de l'argument est supérieure à l'unité
- Méthode utilisée : développement limité de Tchebycheff (voir paragraphe 122 - 511)

231 - 62 Utilisation en virgule flottante

OBJET : Ce sous-programme calcule l'Arc sinus en virgule flottante et simple précision

CONDITIONS D'UTILISATION :

- L'argument est écrit en virgule flottante du type FLAIR (exposant à gauche)
L'unité est alors représentée par : 51 1000 0000 +
- Les limites de l'argument sont en écriture « virgule flottante » : $|\text{Argument}| \leq 51 \ 1000 \ 0000$

Utilisation

Entrée :

- La mémoire d'adresse 1778 doit contenir la constante d'orientation : 69 1757 Δ —
Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1635 ; il est chargé en 1950
Le contenu de 1778 n'est pas modifié par l'exécution du sous-programme .
- 8003 : $\alpha\alpha \ \text{XXXX} \ \text{XXXX} \pm \ (\text{Argument})$
- 8002 : 00 0000 0000
- 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 1750

Sortie :

- 8003 : $\beta\beta \ \text{YYYY} \ \text{YYYY} \pm \ (\text{Arc sinus})$
- 8002 : 00 0000 0000
- 8001 : $\beta\beta \ 0000 \ 0000 +$ si le résultat n'est pas nul
quelconque si le résultat est nul

CARACTERISTIQUES :

- Encombrement : 164 mémoires de Δ à $\Delta + 0098$ et de 1735 à 1799
- Temps de calcul : 345 millisecondes si la valeur absolue de l'argument est inférieure ou égale à $\frac{\sqrt{2}}{2}$
 570 millisecondes si la valeur absolue de l'argument est supérieure à $\frac{\sqrt{2}}{2}$
- Précision : - si la valeur absolue de la mantisse est inférieure ou égale à $\frac{\pi}{4}$, l'erreur sur celle-ci est de $5 \cdot 10^{40 - \beta\beta}$ avec la convention : $0,1 \leq \text{mantisse} < 1$.
 Pour $\beta\beta = 49$ ou 50 , la mantisse est exacte.
 - si la valeur absolue de la mantisse est supérieure à $\frac{\pi}{4}$, l'erreur est de $3 \cdot 10^{-9}$ sauf au voisinage de $\frac{\pi}{2}$ où elle est de $5 \cdot 10^{-8}$, avec la convention : $0,1 \leq \text{mantisse} < 1$
- Arrêt programmé : 01 0015 9999 si $|\text{Argument}| > 51\ 1000\ 0000$
- Méthode utilisée : développement limité de Tchebycheff (voir paragraphe 122 - 511)

232 - Sous-programmes fonctionnels en double précision

Un sous-programme fonctionnel « double précision » n'est utilisable qu'en virgule flottante FLAIR. Il fait appel à l'A.D.P. (arithmétique en virgule flottante double précision) qui doit être enregistrée sur le tambour.

Ce sous-programme se présente sous la forme numérique translatable « 1 mot par carte ». Le facteur de translation Δ qui peut être quelconque, mais limité, est chargé en 1950.

232 - 1 Sous-programmes « Racine Carrée »

Il existe deux sous-programmes « racine carrée » référés DFRCE et DFRCK : le premier est condensé, le second rapide.

232 - 11 Sous-programme DFRCE

OBJET : Ce sous-programme calcule la racine carrée en virgule flottante et double précision, avec un encombrement restreint.

CONDITIONS D'UTILISATION :

- L'argument doit être positif ou nul et écrit en virgule flottante du type FLAIR (exposant à gauche)
 L'unité est alors représentée par : 51 1000 0000 00 0000 0000 +

Utilisation

Entrée :

- 8003 : $\alpha\alpha$ XXXX XXXX +
- 8002 : XX XXXX XXXX +
- 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 0000 + Δ

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1635.
Il est chargé en 1950

Sortie :

- | | |
|-----------------------------------|------------------------------------|
| - 8003 : $\beta\beta$ YYYY YYYY + | - 1700 : 00 0000 00 $\beta\beta$ + |
| - 8002 : YY YYYY YYYY + | - 1715 : 00 YYYY YYYY + |
| - 8001 : quelconque | - 1730 : YY YYYY YYYY + |

REMARQUE : La distance entre les parties gauche et droite d'un nombre en virgule flottante « double précision » reste inchangée après exécution du sous-programme.

CARACTERISTIQUES :

- Encombrement : 65 mémoires de Δ à $\Delta + 0064$ et, obligatoirement, les 300 mémoires (1700 à 1799) de l'A D P
- Temps de calcul : 2,3 secondes
- Précision : l'erreur sur la mantisse est inférieure ou égale à 1.10^{-18} avec la convention : $0,1 \leq \text{mantisse} < 1$
- Arrêt programmé : 01 0021 9999 si l'argument est négatif.
- Méthode utilisée : méthode itérative de Newton (voir paragraphe 111 - 1)

232 - 12 Sous-programme DFRC K

OBJET : Ce sous-programme calcule la racine carrée en virgule flottante et double précision avec un temps de calcul réduit

CONDITIONS D'UTILISATION :

- L'argument doit être positif ou nul et écrit en virgule flottante FLAIR (exposant à gauche)
- L'unité est représentée par : 51 1000 0000 00 0000 0000 +

Utilisation

Entrée :

- 8003 : $\alpha\alpha$ XXXX XXXX +
- 8002 : XX XXXX XXXX +
- 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 0000 + Δ

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1605 .
Il est chargé en 1950 .

Sortie :

-
- 8003 : $\beta\beta$ YYYY YYYY + - 1700 : 00 0000 00 $\beta\beta$ +
 - 8002 : YY YYYY YYYY + - 1715 : 00 YYYY YYYY +
 - 8001 : quelconque - 1730 : YY YYYY YYYY +

REMARQUE : La distance entre les parties gauche et droite d'un nombre en virgule flottante « double précision » reste inchangée après exécution du sous-programme .

CARACTERISTIQUES :

- Encombrement : 95 mémoires de Δ à $\Delta + 0094$ et, obligatoirement, les 300 mémoires (1700 à 1999) de l'A D P
 - Temps de calcul : 0,7 seconde
 - Précision : l'erreur sur la mantisse est inférieure ou égale à 5.10^{-10} avec la convention :
 $0,1 \leq \text{mantisse} < 1$
 - Arrêt programmé : 01 0021 9999 si l'argument est négatif
 - Méthode utilisée : méthode itérative de Newton.
- La première approximation est prise égale à la racine carrée calculée en simple précision

232 - 2 Sous-programme « Logarithme népérien » (Référence DFL E)

OBJET : Ce sous-programme calcule le logarithme népérien en virgule flottante et double précision.

CONDITIONS D'UTILISATION :

- L'argument doit être positif et donné en virgule flottante FLAIR (exposant à gauche)
L'unité est représentée par : 51 1000 0000 00 0000 0000 +
- Les limites de l'exposant de l'argument sont : $08 < \alpha\alpha < 94$

Utilisation

Entrée :

-
- 8003 : $\alpha\alpha$ XXXX XXXX +
 - 8002 : XX XXXX XXXX +
 - 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 0000 + Δ

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1545 .
Il est chargé en 1950 .

Sortie :

- 8003 : $\beta\beta$ YYYY YYYY $\pm$ - 1700 : 00 0000 00 $\beta\beta$ +
- 8002 : YY YYYY YYYY $\pm$ - 1715 : 00 YYYY YYYY $\pm$
- 8001 : quelconque - 1730 : YY YYYY YYYY $\pm$

REMARQUE : La distance entre les parties gauche et droite d'un nombre en virgule flottante « double précision » reste inchangée après exécution du sous-programme .

CARACTERISTIQUES :

- Encombrement : 155 mémoires de Δ à $\Delta + 0154$, et, obligatoirement les 300 mémoires (1700 à 1999) de l'A D P .
- Temps de calcul : 1,4 secondes
- Précision : l'erreur sur la mantisse est inférieure ou égale à $5,10^{32-\beta\beta}$ avec la convention:
 $0,1 \leq \text{mantisse} < 1$
 Pour $\beta\beta$ supérieur à 50, la mantisse est exacte
- Arrêt programmé : 01 0022 9999 si l'argument est négatif ou nul ou si son exposant est en dehors des limites
- Méthode utilisée : développement en série limité et relaxation (voir première partie, paragraphe 122 - 111.2)

232 - 3 Sous-programme « Exponentielle » (référence DFE X)

OBJET : Ce sous-programme calcule l'exponentielle en virgule flottante et double précision.

CONDITIONS D'UTILISATION :

- L'argument doit être écrit en virgule flottante FLAIR (exposant à gauche)
 L'unité est représentée par : 51 1000 0000 00 0000 0000 +
- Les limites de l'argument sont en écriture « virgule flottante » | Argument | < 53 1100 0000 00 0000 0000

Utilisation

Entrée :

- 8003 : $\alpha\alpha$ XXXX XXXX $\pm$
- 8002 : XX XXXX XXXX $\pm$
- 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 0000 + Δ

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1535 .
 Il est chargé en 1950 .

Sortie :

-
- 8003 : $\beta\beta$ YYYY YYYY + - 1700 : 00 0000 00 $\beta\beta$ +
 - 8002 : YY YYYY YYYY + - 1715 : 00 YYYY YYYY +
 - 8001 : quelconque - 1730 : YY YYYY YYYY +

REMARQUE : La distance entre les parties gauche et droite d'un nombre en virgule flottante « double précision » reste inchangée après exécution du sous-programme .

CARACTERISTIQUES :

- Encombrement : 165 mémoires de Δ à $\Delta + 0164$, et, obligatoirement les 300 mémoires (1700 à 1999) de l'A D P
- Temps de calcul : 1,7 secondes
- Précision : l'erreur sur la mantisse est inférieure ou égale à 5.10^{-18} avec la convention : $0,1 \leq \text{mantisse} < 1$
- Arrêt programmé : 01 0023 9999 si l'argument est en dehors des limites imposées
- Méthode utilisée : développement en série limité et relaxation (voir première partie, paragraphe 122 - 211.2)

232 - 4 Sous-programme « Sinus et Cosinus » (Référence D F S C)

OBJET : Ce sous programme calcule le sinus ou le cosinus en virgule flottante et double précision .

CONDITIONS D'UTILISATION :

- L'argument doit être écrit en virgule flottante FLAIR (exposant à gauche)
L'unité est représentée par : 51 1000 0000 00 0000 0000 +
- Les limites de l'argument sont en écriture « virgule flottante »
 $|\text{Argument}| < 54 \ 1000 \ 0000 \ 00 \ 0000 \ 0000$

Utilisation

Entrée :

-
- 8003 : $\alpha\alpha$ XXXX XXXX $\pm$
 - 8002 : XX XXXX XXXX $\pm$
 - 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse :

0000 + Δ pour le calcul du sinus

0001 + Δ pour le calcul du cosinus

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1540 .
Il est chargé en 1950 .

Sortie :

-
- 8003 : $\beta\beta$ YYYY YYYY $\pm$ - 1700 : 00 0000 00 $\beta\beta$ +
 - 8002 : YY YYYY YYYY $\pm$ - 1715 : 00 YYYY YYYY $\pm$
 - 8001 : quelconque - 1730 : YY YYYY YYYY $\pm$

REMARQUE : La distance entre les parties gauche et droite d'un nombre en virgule flottante « double précision » reste inchangée après exécution du sous-programme .

CARACTERISTIQUES :

- Encombrement : 160 mémoires de Δ à $\Delta + 0159$, et, obligatoirement, les 300 mémoires (1700 à 1999) de l' A D P
- Temps de calcul : 1,65 secondes
- Précision : l'erreur sur la mantisse est inférieure ou égale à $2 \cdot 10^{32 - \beta\beta}$ avec la convention:
 $0,1 \leq \text{mantisse} < 1$
- Arrêt programmé : 01 0024 9999 si l'argument est en dehors des limites imposées
- Méthode utilisée : développement limité de Tcheby cheff de $\cos \frac{\pi}{2} X$. (Voir première partie, paragraphe 122 - 311 - 12)

232 - 5 Sous-programme « Arc tangente » (Référence D F A T)

OBJET : Ce sous-programme calcule l'Arc tangente en virgule flottante et double précision

CONDITION D'UTILISATION :

- L'argument doit être donné en virgule flottante FLAIR (exposant à gauche)
- L'unité est représentée par : 51 1000 0000 00 0000 0000 +

Utilisation

Entrée :

-
- 8003 : $\alpha\alpha$ XXXX XXXX $\pm$
 - 8002 : XX XXXX XXXX $\pm$
 - 8001 : Instruction de sortie

La première instruction du sous-programme a pour adresse : 0000 + Δ

Le facteur de translation Δ peut être quelconque pourvu qu'il reste inférieur ou égal à 1505 .
Il est chargé en 1950 .

Sortie :

- 8003 : $\beta\beta$ YYY YYY $\pm$ - 1700 : 00 0000 00 $\beta\beta$ +
- 8002 : YY YYY YYY $\pm$ - 1715 : 00 YYY YYY $\pm$
- 8001 : quelconque - 1730 : YY YYY YYY $\pm$

REMARQUE : La distance entre les parties gauche et droite d'un nombre en virgule flottante double précision reste inchangée après exécution du sous-programme .

CARACTERISTIQUES :

- Encombrement : 195 mémoires de Δ à $\Delta + 0194$, et, obligatoirement, les 300 mémoires (1700 à 1999) de l'A D P
 - Temps de calcul : 1,1 seconde
 - Précision : l'erreur sur la mantisse est inférieure ou égale à $5 \cdot 10^{31-\beta\beta}$ avec la convention:
 $0,1 \leq \text{mantisse} < 1$
 Pour $\beta\beta = 50$ ou 51 , la mantisse est exacte
 - Méthode utilisée : développement en série limité et relaxation (voir première partie,paragraphe 122 - 411.2)
-

CONCLUSION

En comparant les approximations obtenues dans cette étude avec celles que donne HASTINGS, nous remarquons que leur programmation exige davantage de mémoires (30 % environ) ; mais cet inconvénient est largement compensé par une précision meilleure , puisque l'erreur, en virgule fixe et simple précision, par exemple, porte sur le dixième chiffre au lieu du huitième . En outre, le temps de calcul est légèrement réduit (10 % environ) . C'est pourquoi, en dépit d'un encombrement supérieur, l'emploi des approximations traitées ici nous a semblé préférable .

Pour choisir les meilleures d'entre elles, nous avons cherché à remplir au mieux les trois conditions suivantes : un encombrement et un temps de calcul réduits, avec un maximum de précision . Nous avons constaté que, quelle que soit l'approximation, polynomiale ou rationnelle, l'encombrement est sensiblement le même . Pour obtenir une précision aussi grande que possible, nous avons fait, en sorte, de rendre les erreurs de méthodes négligeables devant les erreurs dues aux calculs ; il en résulte que la précision, dépendant alors du matériel utilisé, est la même pour toutes les approximations. Le seul critère pour faire le choix reste donc le temps de calcul.

En général, les approximations rationnelles demandent des calculs beaucoup plus longs puisqu'elles exigent bon nombre de divisions, ce qui les rend mal adaptées au 650 car le temps d'exécution de ces opérations est important sur cet ordinateur.

Parmi les approximations polynomiales, le développement en série avec relaxation est le plus rapide, surtout en double précision. En effet, si le développement en série de Mac - Laurin est moins convergent que le développement de TCHEBYCHEFF, la relaxation permet d'obtenir le même nombre de termes ; en outre, l'emploi des polynômes de TCHEBYCHEFF oblige toujours à se ramener préalablement à une variable pour laquelle l'intervalle de variation soit $(-1,1)$, ce qui exige une multiplication ou une division supplémentaire.

Ainsi le développement en série avec relaxation offre, pour une précision et un encombrement égaux à ceux des autres approximations, un temps de calcul moindre : c'est donc lui qui, en général, satisfait le mieux les conditions requises pour établir les bibliothèques .

A N N E X E S

ANNEXE I

NOTICES EXPLICATIVES DES SOUS-PROGRAMMES FONCTIONNELS
POUR L'ORDINATEUR IBM 650 STANDARD

Dans cette annexe, nous donnons le détail des sous-programmes fonctionnels inclus dans la bibliothèque "Simple Précision".

Directement à la suite, figure le détail d'un sous-programme de "transformation" référé SFVF, sous-programme qui permet d'utiliser la bibliothèque "Simple Précision" indifféremment en virgule fixe ou virgule flottante.

Pour la bibliothèque "Double Précision", nous donnons seulement le détail de DFSC. En effet le passage en Double Précision conserve la logique du programme en Simple Précision; la manipulation de nombres occupant deux mémoires et les conditions particulières d'utilisation du Sous-Programme d'Arithmétique en Virgule Flottante Double Précision sont illustrées par l'étude d'un seul sous-programme et restent identiques pour tous les autres.

La notice explicative de chaque sous-programme se présente sous la forme suivante:

NOTATION

Notation. Au début, nous donnons les notations particulières adoptées pour simplifier l'écriture.

MEMOIRES DE TRAVAIL

Pour une même mémoire, les contenus indiqués sont ceux pris successivement au cours du déroulement du programme. La répétition du signe " = " rappelle un changement du contenu.

```

Par exemple: (0036) = 16 8001 0040 si "A" < 10
               10 0023 0029 si "A" ≥ 10
               = Approximation de Newton.

```

se lit: la mémoire d'adresse 0036 contient d'abord les instructions en langage machine 16 8001 0040 ou 10 0023 0029 suivant la valeur de l'Argument puis par la suite la valeur de l'approximation de Newton.

INSTRUCTIONS

Chaque instruction se présente suivant une ligne décomposée en quatre zones.

Par exemple:

(2) 0030 10 0034 0039 (3) (0034) = π

zone I zone II zone III zone IV

Zone I : Numéro de référence de l'instruction correspondante, chaque fois qu'elle sera exécutée plusieurs fois.

Zone II : Adresse de l'instruction à étudier et instruction en langage machine.

Zone III : Numéro de référence de l'instruction à suivre si elle n'est pas étudiée à la ligne suivante.
Remarque: en cas de test ce numéro peut se trouver sous l'adresse facteur.

Zone IV : Commentaires.

REPRESENTATION D'UNE SEQUENCE D'INSTRUCTIONS

XXXX

Opération réalisée
 par cette séquence
 YYYY

XXXX : Adresse de la première instruction de la séquence.

YYYY : Adresse de l'instruction à exécuter dès la fin du déroulement de la séquence.

Cette représentation a lieu chaque fois que :

- la séquence, vue son importance, fait l'objet d'une étude séparée.

Par exemple, dans le sous-programme SFVF :

△

Calcul de la fonction
 en virgule fixe.

1757

- la séquence est reprise plusieurs fois au cours du déroulement du sous-programme. Le détail de cette séquence est alors donné en fin de l'étude.

Par exemple, dans le sous-programme DFSC :

0122

Préparation de l'entrée
 dans l'A.D.P pour effectuer une multiplication.

1806

- la séquence n'est pas étudiée parce qu'elle fait partie d'un sous-programme auxiliaire.

Par exemple, dans le sous-programme DFSC :

1806

A.D.P
 Multiplication des
 mantisses.

1821

Remarque 1.

Pour une étude complète du sous-programme A.D.P, nous renvoyons à la bibliographie (Annexe II, 7)

Remarque 2.

Comme tous les calculs sont effectués en virgule fixe, nous donnons après chaque opération la position de la virgule fictive, en indiquant le cadrage du résultat. Les cadrages sont toujours comptés à partir de la position extrême gauche de la mémoire contenant la donnée.

Par exemple, si 8000 est considéré comme une mémoire de 20 positions, réunissant les deux accumulateurs:

$$(8000) = A \text{ cadré } 11$$

La virgule fictive de l'argument est placée après la 11^e position comptée à partir de la position 20 des accumulateurs.

Notation

A : Argument cadré II dans l'accumulateur à compter de la position gauche et limité à IQ chiffres significatifs.

a : Argument réduit dans l'intervalle (I, IO).

La racine est mise sous la forme : $\sqrt{A} = \sqrt{a} \cdot IO^{\frac{n}{2}}$

Si "m" est positif (A supérieur ou égal à IO), nous posons : $\frac{m}{2} = 5 - \frac{n}{2}$

Si "m" est négatif ou nul (A inférieur à IO), nous posons : $\frac{m}{2} = -\frac{n}{2}$

"n" est le nombre de décalages à gauche pour obtenir "a" cadré I

$$n = 2k + I$$

$$\begin{aligned} \frac{m}{2} &= 5 - k - \frac{I}{2} & \sqrt{A} &= \sqrt{IOa} \cdot IO^{\frac{4-k}{2}} \\ \frac{m}{2} &= -k - \frac{I}{2} & \sqrt{A} &= \sqrt{IOa} \cdot IO^{-\frac{(k+1)}{2}} \end{aligned}$$

$$n = 2k$$

$$\begin{aligned} \frac{m}{2} &= 5 - k & \sqrt{A} &= \sqrt{a} \cdot IO^{\frac{5-k}{2}} \\ \frac{m}{2} &= -k & \sqrt{A} &= \sqrt{a} \cdot IO^{-\frac{k}{2}} \end{aligned}$$

"IOa" ou "a" est cadré 2. $\sqrt{IOa}$ ou $\sqrt{a}$ est cadré I.

La racine transférée en 8002 donne le cadrage final II. La puissance de IO du résultat est enregistrée dans la position unité de 800I, avec son signe.

La première instruction du sous-programme se trouve en 0000

Mémoires de travail

à contenus constants

(0023) = 00 0000 0005

(0026) = 00 0000 0002

(003I) = 50 0000 0000

(004I) = 99 9999 9999

à contenus variables

(0007) = Instruction de sortie du sous-programme.

(00I6) = "a" cadré I.

= "IOa" ou "a" cadré 2.

(0034) = x0 0000 0000 si le calcul porte sur "a". ("x" est le dixième chiffre de "a" cadré 2).

= 00 0000 0000 si le calcul porte sur "IOa".

(0036) = Mot d'orientation

son contenu est :

I6 800I 0040 si "A" est inférieur à IO.

IO 0023 0029 si "A" est supérieur ou égal à IO.

= Approximation de Newton dans les itérations.

(0044) = Puissance de IO du résultat :

00 0000 000(4-k) si "m" est positif et impair.

00 0000 000(5-k) si "m" est positif et pair.

-00 0000 000(k+I) si "m" est négatif et impair.

-00 0000 000(k) si "m" est négatif et pair.

00 0000 0000 si "m" est nul.

Instructions

0000 46 0003 0004 (I)

0003 0I 00II 9999

(I)0004 24 0007 00IO Arrêt programmé pour un argument négatif.
(0007) = Instruction de sortie.

00IO 45 00I4 00I5

(2)

00I5 60 8002 0007

Sortie pour un argument nul.

(La seule utilité de cette instruction est d'annuler le contenu de 800I afin d'obtenir, même dans ce cas, la puissance de IO du résultat.)

(2)00I4 44 00I7 00I8

(3)

00I8 60 8002 0027

0027 69 0030 0033 (4)

(0030) = I6 800I 0040

L'argument est inférieur à IO. Enregistrement du mot d'orientation correspondant dans (0036).

(3)00I7 69 0020 0033 (0020) = IO 0023 0029

L'argument est supérieur ou égal à IO. Enregistrement du mot d'orientation dans (0036).

(4)0033 24 0036 0039

0039 36 0000 00II

00II 2I 00I6 00I9

(00I6) = "a" cadré I.

00I9 6I 8002 0025

(8002) = 00 0000 000m

0028 I9 003I 0036

(003I) = 50 0000 0000

(8003) = 00 0000 000k

(8002) = 00 0000 0000 si "n" est pair.

=-50 0000 0000 si "n" est impair

Calcul de la puissance de IO du résultat et stockage dans la mémoire (0044). 8002 contient alors un mot de contrôle permettant de tester la parité de "n" pour former "a" ou "IOa" suivant les cas.

(0036) = Mot d'orientation : I6 800I 0040

(800I) = 50 0000 0000

La soustraction de 5 en position IO de

8002 :

-entraîne un report d'unité sur 8003

si "n" est impair.

(8003) = 00 0000 000(k+I)

(8002) = 00 0000 0000

-n'entraîne aucune modification de 8003

si "n" est pair.

(8003) = 00 0000 000(k)

(8002) = 00 0000 0000

0040 2I 0044 0047 (5)

0020 IO 0023 0029

(0036) = Mot d'orientation: IO 0023 0029

(0023) = 00 0000 0005

La partie entière contenue en position

I de 8003 est diminuée d'une unité si

"n" est impair :

(8003) = 00 0000 000(4-k)

(8002) = 50 0000 0000

"n" est pair :

(8003) = 00 0000 000(5-k)

(8002) = 00 0000 0000

0029 2I 0044 0048

0048 I5 003I 0047

(003I) = 50 0000 0000

Le mot de contrôle contenu en 8002 est:

-nul si "n" est impair.

-non nul si "n" est pair.

(5) 0047 60 8002 0005

0005 44 0009 00I2 (6)

0009 60 00I6 002I

002I 30 000I 0032 (7)

(00I6) = "a" cadré I.

"a" est cadré 2 (n pair) et par suite son stockage requiert deux mémoires pour conserver tous les chiffres significatifs.

"n" est impair. "a" cadré I est égal à

"IOa" cadré 2. Le contenu de (00I6) reste inchangé.

(7) 0032 2I 00I6 0022

0022 20 0034 0037

0037 II 004I 0045

0045 44 006I 0002 (8)

(00I6) et (0034) = "a" ou "IOa" cadré 2.

(004I) = 99 9999 9999

Renvoi à la sortie de la boucle si

IOa = 99 9999 9999

000I 65 800I 00I3

Calcul de $\sqrt{a}$ ou $\sqrt{IOa}$ par la méthode itérative de Newton avec pour première approximation $X_N = 99\ 9999\ 9999$. Il y a sortie de la boucle (9) dès que $\frac{a}{X_N}$ ou $\frac{IOa}{X_N}$ est supérieur ou égal à X_N .

(9) 00I3 20 0036 0042 (0036) = Approximation de Newton.

0042 60 00I8 0024

0024 I5 0034 0043

0043 64 0036 0006

0006 I6 800I 0025

0025 46 0035 0002 (8)

0035 I5 800I 0046

0046 I5 800I 0008

0008 64 0026 00I3 (9)

(0026) = 00 0000 0002

(800I) = $\sqrt{a}$ ou $\sqrt{IOa}$ cadré I.

(0044) = Puissance de IO du résultat.

(0007) = Instruction de sortie.

Constantes pour le calcul de la première approximation I_p .

Notation

Sous-programme SFRC.K

A : Argument cadré 11 dans l'accumulateur à compter de la position gauche et limité à 10 chiffres significatifs.

a : Argument réduit dans l'intervalle (0,1) 11.

La racine est mise sous la forme : $A = \sqrt{10a} \cdot 10^{\frac{n}{2}}$

Si "m" est positif (A supérieur ou égal à 10), nous posons : $\frac{m}{2} = 5 - \frac{n}{2}$

Si "m" est négatif ou nul (A inférieur à 10), nous posons : $\frac{m}{2} = -\frac{n}{2}$

"n" est le nombre de décalages à gauche pour obtenir "a" cadré 0.

$$n = 2k + 1$$

$$\frac{m}{2} = 5 - k - \frac{1}{2} \quad \sqrt{A} = 10 \sqrt{a} \cdot 10^{4-k}$$

$$\frac{m}{2} = -k - \frac{1}{2} \quad \sqrt{A} = 10 \sqrt{a} \cdot 10^{-(k+1)}$$

$$n = 2k$$

$$\frac{m}{2} = 5 - k \quad \sqrt{A} = 10^{\frac{1}{2}} \sqrt{a} \cdot 10^{5-k}$$

$$\frac{m}{2} = -k \quad \sqrt{A} = 10^{\frac{1}{2}} \sqrt{a} \cdot 10^{-k}$$

"a" est localisé dans l'un des trois sous-intervalles suivants :

$$I_1(0,1 \ 0,23) \quad I_2(0,23 \ 0,529) \quad I_3(0,529 \ 1)$$

pour permettre le choix des constantes A_p, B_p, C_p, D_p ($p = 1, 2, 3$) figurent dans la première approximation I_p .

$10 \sqrt{a}$ ou $10^{\frac{1}{2}} \sqrt{a}$ est cadré 11 dans l'accumulateur. La puissance de 10 du résultat est enregistrée dans la position unité de 8001, avec son signe.

La première instruction du sous-programme se trouve en 0000.

Mémoires de travail

à contenus constants

$$(0023) = 00 \ 0000 \ 0005$$

$$(0024) = 10 \ 0000 \ 0000$$

$$(0035) = 10 \ 0000 \ 0001$$

$$(0046) = 99 \ 9999 \ 9999$$

$$(0080) = 50 \ 0000 \ 0000$$

$$(0086) = 31 \ 6227 \ 7660 \ (10^{\frac{1}{2}})$$

$$(0103) = 00 \ 0000 \ 0002$$

Constantes pour la localisation de "a" dans I_p :

$$(0022) = 02 \ 9900 \ 0000 \ (0,529 - 0,23) \quad (0061) = 05 \ 2900 \ 0000$$

$$(0034) = 62 \ 1722 \ 6200 \ (A_1)$$

$$(0064) = 08 \ 0516 \ 4872 \ (B_1)$$

$$(0052) = 01 \ 0244 \ 1995 \ (C_1)$$

$$(0095) = 01 \ 5836 \ 5577 \ (D_1)$$

$$(0085) = 40 \ 9951 \ 7550 \ (A_2)$$

$$(0063) = 18 \ 5187 \ 9206 \ (B_2)$$

$$(0002) = 05 \ 4191 \ 8154 \ (C_2)$$

$$(0045) = 03 \ 6424 \ 0827 \ (D_2)$$

$$(0073) = 27 \ 0314 \ 1820 \ (A_3)$$

$$(0059) = 42 \ 5932 \ 2172 \ (B_3)$$

$$(0051) = 28 \ 6674 \ 7032 \ (C_3)$$

$$(0093) = 08 \ 3775 \ 3902 \ (D_3)$$

à contenus variables

(0007) = Instruction de sortie du sous-programme.

(0016) = "a" cadré 0.

(0036) = Mot d'orientation

son contenu est :

16 8001 0089 si "A" est inférieur à 10.

10 0023 0028 si "A" est supérieur ou égal à 10.

= A_p cadré 0 ($p = 1, 2, 3$)

= Première approximation I_p cadré 0

(0044) = Puissance de 10 du résultat :

00 0000 000(4-k) si "m" est positif et impair.

00 0000 000(5-k) si "m" est positif et pair.

-00 0000 000(k+1) si "m" est négatif et impair.

-00 0000 000(k) si "m" est négatif et pair.

00 0000 0000 si "m" est nul.

(0055) = "a" cadré 1 et arrondi à 9 chiffres significatifs. Le contenu de cette mémoire est utilisé dans la localisation de "a" et dans le calcul de la première approximation.

(0067) = B_p cadré 1 ($p = 1, 2, 3$)

(0068) = Mot d'orientation

(0068) = Mot d'orientation
 son contenu est :
 60 8002 0054 si $\sqrt{a}$ est multiplié par 10^4 .
 69 0044 0007 si $\sqrt{a}$ est multiplié par 10.
 (0079) = a+Dp cadré I (p = 1,2,3)

Instructions

0000	46	0003	0004	(1)	
0003	01	0011	9999		Arrêt programmé pour un argument négatif.
(1)	0004	24	0007	0010	(0007) = Instruction de sortie.
0010	45	0014	0015	(2)	
0015	60	8003	0007		Sortie pour un argument nul.
					La seule utilité de cette instruction est d'annuler le contenu de 8001 afin d'obtenir, même dans ce cas, la puissance de 10 du résultat en position unité.
(2)	0014	44	0017	0018	
	0018	60	8002	0027	(3)
0027	69	0030	0033	(4)	(0030) = I6 8001 0089
					L'argument est inférieur à 10. Enregistrement du mot d'orientation dans (0036)
(3)	0017	69	0020	0033	(0020) = 10 0023 0028
					L'argument est supérieur ou égal à 10. Enregistrement du mot d'orientation correspondant dans (0036).
(4)	0033	24	0036	0039	
0039	36	0000	0011		(0016) = "a" cadré 0.
0011	21	0016	0019		(8002) = 00 0000 000n
0019	61	8002	0077		(0080) = 50 0000 0000
0077	19	0080	0036		(8003) = -00 0000 000k
					(8002) = -00 0000 0000 si "n" est pair.
					= -50 0000 0000 "n" impair.

Calcul de la puissance de 10 du résultat et stockage dans la mémoire (0044). 8002 contient alors un mot de contrôle permettant de tester la parité de "n" afin de pouvoir enregistrer le mot d'orientation correspondant dans la mémoire (0068).

0030	I6	8001	0089	(0036) = Mot d'orientation: I6 8001 0089	
				(8001) = 50 0000 0000	
				La soustraction de 5 en position 10 de 8002 :	
				-entraîne un report d'unité sur 8003	
				si "n" est impair :	
				(8003) = -00 0000 000(k+1)	
				(8002) = -00 0000 0000	
				-n'entraîne aucune modification de 8003	
				si "n" est pair :	
				(8003) = -00 0000 000k	
				(8002) = -50 0000 0000	
0089	21	0044	0047	(5)	
				(0036) = Mot d'orientation: 10 0023 0028	
0020	10	0023	0028	(0023) = 00 0000 0005	
				La partie entière contenue en position 1 de 8003 est diminuée d'une unité si "n" est impair :	
				(8003) = 00 0000 000(4-k)	
				(8002) = 50 0000 0000	
				Si "n" est pair :	
				(8003) = 00 0000 000(5-k)	
				(8002) = 00 0000 0000	
0028	21	0044	0097		
0097	15	0080	0047	(0080) = 50 0000 0000	
				Le mot de contrôle contenu en 8002 est nul si "n" est impair, non nul si "n" est pair.	
(5)	0047	60	8002	0005	
	0005	44	0009	0060	(6)
	0009	69	0012	0065	(7)
				(0012) = 60 8002 0054	

(6) 0060	69 0013 0065	"n" est pair. Enregistrement du mot d'orientation correspondant dans la mémoire (0068). (0013) = 69 0044 0007	(10)0069	15 0022 0078	(0022) = 02 9900 0000
			0078	46 0031 0082 (12)	
			0082	69 0085 0083	(0085) = A ₂ cadré 0.
			0083	24 0036 0041	
(7) 0065	24 0068 0021	"n" est impair. Enregistrement du mot d'orientation dans la mémoire (0068). (0016) = "a" cadré 0. (0024) = 10 0000 0000	0041	60 0045 0049	(0045) = D ₂ cadré I.
			0049	15 0002 0057	(0002) = C ₂ cadré I.
			0057	69 0063 0062 (II)	(0063) = B ₂ cadré I.
			(12)0031	69 0034 0037	(0034) = A ₁ cadré 0.
	0029 45 0032 0053 (8)	Résolution du cas où A = 10 ⁹ (q entier positif, négatif ou nul). $\sqrt{a}$ égale 10 ⁻⁴ qui est alors enregistré en 8002, cadré 0. Il y a renvoi à (0068). (0086) = 31 6227 7660 (0035) = 10 0000 0001	0037	24 0036 0091	
			0091	60 0095 0099	(0095) = D ₄ cadré I.
			0099	15 0052 0058	(0052) = C ₄ cadré I.
			0058	69 0064 0062	(0064) = B ₄ cadré I.
(8) 0032	15 0035 0040	Il y a report d'unité sur 8003 si "a" est égal à 99 9999 9999. 0040 44 0043 0094 (9) Résolution du cas où "a" = 99 9999 9999. $\sqrt{a}$ a même valeur. Après l'enregistrement de cette valeur en 8002, il y a renvoi à (0068). (0046) = 99 9999 9999	(II)0062	24 0067 0072	(0067) = B _p cadré I.
			calcul de la première approximation $X_p = A_p (B_p + a - \frac{C_p}{a^2 B_p})$		
			0072	10 0055 0074	(0055) = "a" cadré I.
			0074	21 0079 0084	(0079) = a + D _p cadré I.
(9) 0094	31 0001 0001	Localisation de "a" dans I _p et enregistrement des constantes correspondantes. (0061) = 05 2900 0000 (0073) = A ₃ cadré 0. (0093) = D ₃ cadré I. (0051) = C ₃ cadré I. (0059) = B ₃ cadré I.	0084	61 8002 0086	(8000) = -C _p cadré 2.
			0086	30 0001 0053	
			0053	64 0079 0042	
			0042	15 0055 0025	
	0043 65 0046 0068	fres significatifs. Localisation de "a" dans I _p et enregistrement des constantes correspondantes. (0061) = 05 2900 0000 (0073) = A ₃ cadré 0. (0093) = D ₃ cadré I. (0051) = C ₃ cadré I. (0059) = B ₃ cadré I.	0025	15 0067 0075	(0067) = B _p cadré I.
			0075	60 8002 0087	
			0087	19 0036 0076	(0036) = A _p cadré 0.
			0076	65 8003 0088	(8003) = X _p cadré I.
	0026 24 0036 0090	Dans le cas où X _p est supérieur ou égal à I, il est remplacé par 99 9999 9999 afin de conserver le cadrage 0 pour la première approximation et par suite pour $\sqrt{a}$. (0036) = Première approximation cadré 0.	0088	35 0001 0098	
			0098	44 0101 0102 (I3)	
			0101	65 0046 0102	
			(I3)0102	20 0036 0092	

Calcul de $\sqrt{a} = \frac{1}{2} \left(\frac{s}{x_p} + x_p \right)$

0092 60 0018 0081

(0018) = "a" cadré 0, avec les 10 chiffres significatifs initiaux.

La valeur maximale de "a" est 9999999998 et elle est toujours inférieur à celle de x_p .

0081 64 0036 0050

0050 15 8001 0100

0100 64 0103 0068

(0103) = 00 0000 0002

(0068) = Mot d'orientation: 60 8002 0054

0012 60 8002 0054

0054 19 0036 0104

0104 15 0080 0056

(0086) = $10^{\frac{1}{2}}$

(0080) = 50 0000 0000

Le résultat $10^{\frac{1}{2}}\sqrt{a}$ est arrondi par report ou non d'unité.

0056 65 8003 0013

Cadrage du résultat; II dans 8000.

(0068) = Mot d'orientation: 69 0044 0007

$\sqrt{a}$ est cadré 0 en 8002. $10\sqrt{a}$ est égal au contenu de 8002, cadré I.

0013 69 0044 0007

(0044) = Puissance de 10 du résultat.

(0007) = Instruction de sortie.

Sous-programme SFLK

Notation

A : Argument cadré 11 dans l'accumulateur à compter de la position gauche et limité à 10 chiffres significatifs.

a_1 : Argument réduit dans l'intervalle $(1, 10^{\frac{4}{2^i}})$

Détermination de a_1

$A = a_0 \cdot 10^m$ avec $1 \leq a_0 < 10$

Si "m" est positif (A supérieur ou égal à 10), nous posons : $m = 10 - n$

Si "m" est négatif ou nul (A inférieur à 10), nous posons : $m = -n$

"n" est le nombre de décalages à gauche pour obtenir a_0 cadré 1.

$a_{1-4} = a_1 \cdot 10^{\frac{4-i}{2^i}}$ avec $1 \leq a_1 < 10^{\frac{4}{2^i}}$

" ε_i " = 1 si a_{1-4} est supérieur ou égal à $10^{\frac{4}{2^i}}$

" ε_i " = 0 si a_{1-4} est inférieur à $10^{\frac{4}{2^i}}$

Nous limitons la réduction à $i = 3$ et A se présente sous la forme :

$$A = a_3 \cdot 10^{(m + \sum_{i=1}^3 \frac{\varepsilon_i}{2^i})} \quad \text{avec } 1 \leq a_3 < 10^{\frac{4}{2^3}}$$

Soit : $\text{Log}_2 A = \text{Log}_2 a_3 + (m + \sum_{i=1}^3 \frac{\varepsilon_i}{2^i}) \text{Log}_2 10$

Changement de variable

$$t = \frac{a_3 \cdot 10^{\frac{4}{2^i}}}{a_3 + 10^{\frac{4}{2^i}}} \quad \text{Log}_2 A = \text{Log}_2 \frac{1+t}{1-t} + (m + \frac{1}{2^i} + \sum_{i=1}^3 \frac{\varepsilon_i}{2^i}) \text{Log}_2 10$$

$\text{Log}_2 A$ est obtenu cadré 11 dans l'accumulateur.

La première instruction du sous-programme se trouve en 0000.

Mémoires de travail

à contenus constants

Constantes pour la réduction d'intervalle :

(0012) = 10 0000 0000

(0068) = 00 0000 0010

(0086) = 00 0000 5000

(0087) = 00 0000 2500

(0038) = 00 0000 1250

(0088) = 00 0000 0625

(0032) = 31 6227 7660 $(10^{\frac{4}{2^3}})$

(0062) = 17 7827 9410 $(10^{\frac{4}{2^2}})$

(0013) = 13 3352 1432 $(10^{\frac{4}{2^1}})$

(0063) = 11 5478 1985 $(10^{\frac{4}{2^0}})$

(0076) = 23 0258 5093 $(\text{Log}_2 10)$

Coefficients pour le calcul de $\text{Log}_2 \frac{1+t}{1-t}$

(0077) = 20 0000 0004 (K_1)

(0006) = 66 6659 9482 (K_3)

(0055) = 40 2592 0000 (K_5)

à contenus variables

(0011) = Instruction de sortie du sous-programme.

(0026) = $10^{\frac{4}{2^i}}$ cadré 1 ($i=1,2,3$) dans la réduction d'intervalle.

= t^2 cadré 0 dans le calcul de $\text{Log}_2 \frac{1+t}{1-t}$

(0030) = $m + \sum_{i=1}^3 \frac{\varepsilon_i}{2^i} + \frac{1}{2^i}$, cadré 6. Le contenu de cette mémoire varie au cours de la réduction d'intervalle.

(0054) = " a_1 " cadré 1 ($i=0,1,2,3$). " i " varie d'une unité à chaque réduction.

= t cadré 0 dans le calcul de $\text{Log}_2 \frac{1+t}{1-t}$

(0089) = Instruction à exécuter en fin de la réduction " i ".

Son contenu est :

($i=1$) 10 0086 0042 addition de $\frac{1}{2}$ au contenu de 0030

($i=2$) 10 0087 0043 " $\frac{1}{4}$ " "

($i=3$) 10 0038 0044 " $\frac{1}{8}$ " "

= $a_3 + 10^{\frac{4}{2^i}}$ cadré 1 dans le calcul de " t ".

(0026) = $\text{Log}_2 \frac{1+t}{1-t}$ cadré 1.

Instructions

0000 46 0003 0004

(1)

0004 45 0008 0003

(2)

(1) 0003 01 0012 9999

Arrêt programmé pour un argument négatif ou nul.

(2) 0003 24 0011 0014

(0011) = Instruction de sortie.

0014 44 0017 0018
(3)Détermination de a_0 . Calcul de "m" et stockage dans la mémoire (0030)

0018 60 8002 0027

0027 36 0000 0049

0049 21 0054 0007 (0054) = a_0 cadré 1.

0007 61 8002 0015 (4) (8002) = 00 0000 000n

(3) 0017 36 0000 0039

0039 21 0054 0057 (0054) = a_0 cadré 1.

0057 61 8002 0065 (8002) = 00 0000 000n

0065 10 0068 0015 (0068) = 00 0000 0010

(4) 0015 35 0004 0025

0025 21 0030 0033 (0030) = "m" cadré 6.

Résolution du cas où $A = 10^m$

0033 60 0054 0009

0009 11 0012 0067 (0012) = 10 0000 0000

0067 45 0020 0025 (5) $\log_{10} \frac{A+E}{1-E}$ est nul. Renvoi à la fin du calcul de $\log_{10} \frac{A+E}{1-E}$ Réduction de " a_0 " dans l'intervalle $(1, 10^{\frac{1}{6}})$. Calcul de la quantité $(m + \sum_{i=1}^5 \frac{E_i}{2^i} + \frac{1}{16})$ et stockage en (0030).

0020 10 8001 0079

0079 11 0032 0037 (0032) = $10^{\frac{1}{2}}$

0037 46 0040 0041 (6)

" a_0 " est supérieur ou égal à $10^{\frac{1}{2}}$. Enregistrement en (0069) de l'instruction à exécuter après la réduction et renvoi à la séquence (7) qui opère la réduction de " a_{i-1} " en " a_i " (ici $i=1$). A la sortie de cette séquence, il y a addition de $\frac{1}{2}$ à "m".

0041 24 0026 0080 (0026) = $10^{\frac{1}{2}}$

0080 69 0083 0037 (7) (0083) = 10 0086 0042

Instruction qui sera exécutée après la réduction.

(8) 0083 10 0086 0042

0042 21 0030 0040 (0030) = $m + \frac{1}{2}$ cadré 6.

La réduction ($i=1$) est terminée et le contenu de (0030) a été modifié en conséquence ou elle n'a pas eu lieu (" a_0 " inférieur à $10^{\frac{1}{2}}$) et le contenu de (0030) est resté inchangé. Dans les deux cas nous avons " a_1 " appartenant à l'intervalle $(1, 10^{\frac{1}{2}})$ et la mémoire (0030) convenablement chargée. Nous passons à la réduction suivante.

(6) 0040 60 0054 0059 (0054) = $a_1 = a_0$ si " a_0 " inférieur à $10^{\frac{1}{2}}$
 $= a_0 \cdot 10^{-\frac{1}{2}}$ s'il y a eu passage par (7).

0059 11 0062 0019 (0062) = $10^{\frac{1}{4}}$

0019 46 0022 0023 (9)

" a_1 " est supérieur ou égal à $10^{\frac{1}{4}}$. Exécution de la réduction ($i=2$).

0023 24 0026 0031 (0026) = $10^{\frac{1}{4}}$

0031 69 0034 0036 (7) (0034) = 10 0087 0043

Instruction qui sera exécutée après la réduction. (Modification du contenu de la mémoire 0030).

(10) 0034 10 0087 0043 (0087) = 00 0000 2500

0043 21 0030 0022 (0030) = $m + \frac{E_1}{2} + \frac{1}{4}$, cadré 6.

" E_1 " = 1 ou 0 suivant que la réduction ($i=1$) a eu lieu ou non.

La réduction ($i=2$) est terminée ou elle n'a pas eu lieu (" a_1 " inférieur à $10^{\frac{1}{4}}$). Nous passons à la dernière réduction.

(9) 0022 60 0054 0010 (0054) = a_2 0010 11 0013 0069 (0013) = $10^{\frac{1}{8}}$

0069 46 0072 0073 (11)

" a_2 " est supérieur ou égal à $10^{\frac{1}{8}}$. Exécution de la réduction ($i=3$)

0073 24 0026 0081 (0026) = $10^{\frac{1}{8}}$

0081 69 0084 0036 (7) (0084) = 10 0038 0044

Modification de 0030, après réduction.

(12) 0084 10 0038 0044 (0038) = 00 0000 1250

0044 21 0030 0072 (11) (0030) = $m + \frac{E_1}{2} + \frac{E_2}{4} + \frac{1}{8}$, cadré 6.

" E_2 " = 1 ou 0 suivant que la réduction ($i=2$) a été nécessaire ou non.

Réduction de "a_{i-4}" en "a_i".

(7) 0036 24 0089 0045

(0089) = Instruction à exécuter à la

sortie de cette séquence.

0045 10 0026 0082

(0026) = 10^{1/4} (i=1,2,3)

Cette addition reforme "a_{i-4}" dont la valeur avait été modifiée par suite de sa comparaison avec 10^{1/4}.

0082 30 0001 0090

(8000) = a_{i-4} cadré 2.

0090 64 8001 0005

(8001) = 10^{1/2}

0005 20 0054 0058

(0054) = a_i cadré 1.

0058 60 0030 0089

Rappel du contenu de la mémoire (0030) pour le modifier.

(8) pour i = 1.

(10) pour i = 2.

(12) pour i = 3.

La réduction est terminée. Addition de $\frac{1}{16}$ au contenu de (0030). Calcul de la nouvelle variable "t" et de son carré.

(11) 0072 60 0030 0035

0035 10 0088 0046

(0088) = 00 0000 0625

0046 21 0030 0085

(0030) = $m + \frac{1}{16} + \sum_{i=1}^3 \frac{e_i}{2^i}$, cadré 6.

0085 60 0054 0060

(0054) = a₃ cadré 1.

0060 10 0063 0070

(0063) = 10^{1/16}

0070 21 0089 0047

0047 60 0054 0061

0061 11 0063 0071

0071 64 0089 0050

0050 60 8002 0064

(0054) = t cadré 0.

0064 24 0054 0016

(0026) = t² cadré 0.

0016 19 8001 0048

0048 21 0026 0001

Calcul de Log_e $\frac{1+t}{1-t}$

Log_eA.

0001 60 0055 0066

(0055) = K₅ cadré 0.

0066 19 0026 0051

0051 10 0006 0024

(0006) = K₃ cadré 0.

0024 60 8003 0002

0002 19 0026 0052

Cadrage 1 pour l'addition de K₄.

0052 30 0001 0074

(0077) = K₄ cadré 1.

0074 10 0077 0053

0053 60 8003 0075

0075 19 0054 0021

(5) 0021 21 0026 0029
(0026) = Log_e $\frac{1+t}{1-t}$ cadré 1 si A est différent de 10^m.

= 00 0000 0000 si A = 10^m.

(0030) = $m + \frac{1}{16} + \sum_{i=1}^3 \frac{e_i}{2^i}$, cadré 6. (A ≠ 10^m)
= m si A = 10^m.

(0076) = Log₁₀ cadré 1.

Cadrage 11 dans l'accumulateur. (Le décalage ne se fait pas avec arrondi car Log₁₀ est arrondi supérieurement.)

0078 15 0026 0011

(8000) = Log_eA cadré 11.

(0011) = Instruction de sortie.

Sous-programme SFEX.E

Notation

A : Argument cadré 11 dans l'accumulateur à compter de la position gauche. A peut avoir jusqu'à 19 chiffres significatifs.

a : Argument réduit dans l'intervalle $(-\frac{1}{2} \text{Log}_e 10, \frac{1}{2} \text{Log}_e 10)$

Suivant son signe, A est décomposé sous la forme :

$$A = \left[N + \frac{1}{2} + (F - \frac{1}{2}) \right] \cdot \text{Log}_e 10 \quad \text{si } A \text{ est positif.}$$

$$A = \left[- (N+1) + \frac{1}{2} + (\frac{1}{2} - F) \right] \cdot \text{Log}_e 10 \quad \text{si } A \text{ est négatif.}$$

"N" est entier positif. "F" est un nombre fractionnaire positif. $(0 \leq F < 1)$

L'exponentielle devient : $e^A = 10^m \cdot 10^{\frac{1}{2}} \cdot e^a$

avec "m" = N et $a = (F - \frac{1}{2}) \cdot \text{Log}_e 10$ si A est positif.

avec "m" = - (N+1) et $a = (\frac{1}{2} - F) \cdot \text{Log}_e 10$ si A est négatif.

e^a est obtenu par une approximation rationnelle de forme : $\frac{\sum_{n=0}^N K_n (a)^n}{\sum_{n=0}^N K_n (-a)^n}$

Il est ensuite multiplié par $10^{\frac{1}{2}}$. Le résultat est cadré 11 dans l'accumulateur.

"m" : puissance entière de 10 du résultat est enregistrée en 8001, avec son signe. (Le chiffre unité est en position 1 du distributeur).

La première instruction du sous-programme se trouve en 0000.

Mémoires de travail

à contenus constants

- (0014) = 10 0000 0000
 (0029) = 00 0000 0001
 (0013) = 23 0258 5093 ($\text{Log}_e 10$)
 (0050) = 11 5129 2546 ($\frac{1}{2} \text{Log}_e 10$)
 (0062) = 31 6227 7660 ($10^{\frac{1}{2}}$)

Coefficients de l'approximation rationnelle.

- (0055) = 50 0000 0000 (K_1)
 (0002) = 11 3636 3636 (K_2)
 (0046) = 01 5151 5151 (K_3)
 (0042) = 00 1262 6263 (K_4)
 (0043) = 00 0063 1313 (K_5)
 (0017) = 00 0001 5031 (K_6)

à contenus variables

- (0003) = Instruction de sortie du sous-programme.
 (0022) = Dénominateur de l'approximation rationnelle.
 (0024) = Instruction à exécuter en fin de la séquence (4).

Cette séquence est reprise deux fois : une première fois pour le calcul du dénominateur, une seconde fois pour le calcul du numérateur de l'approximation rationnelle.

Son contenu est :

21 0022 0025 au premier passage.

30 0001 0004 au second passage.

(0037) = Puissance de 10 du résultat, cadré 10. (Mlle peut atteindre 10 chiffres significatifs.)

(0060) = "a" cadré 1, dans le calcul du dénominateur.

= "a" cadré 1, dans le calcul du numérateur.

Instructions

0000 24 0003 0006 (0003) = Instruction de sortie.

0006 45 0010 0011
 (1)

L'argument est nul. Enregistrement de 1 en 8002 et de 0 en 8001.

0011 15 0014 0019 (0014) = 10 0000 0000

0019 69 8003 0003 Sortie pour un argument nul.

Calcul de "a" et de la puissance de 10 du résultat qui sera stockée en (0037).

(1) 0010 14 0013 0023 (0013) = $\text{Log}_e 10$

0023 46 0026 0027 (2)

A est négatif.

(8002) = -N cadré 10.

(8003) = -F. $\text{Log}_e 10$ cadré 1.

(0029) = 00 0000 0001

(0037) = - (N+1) cadré 10.

0040 60 8003 0047

0047 10 0050 0005 (3) (0050) = $\frac{1}{2} \text{Log}_e 10$

A est positif.

(8002) = N cadré 10.

(8003) = F.Log₁₀ cadré 1.

(0037) = N cadré 10.

0027 20 0037 0041

0041 60 8003 0049

0049 11 0050 0005

(3) 0005 21 0060 0063

(0050) = $\frac{4}{x} \text{Log}_{10}$

(0060) = $(F - \frac{1}{2}) \text{Log}_{10}$ si A est positif

= $(\frac{1}{2} - F) \text{Log}_{10}$ si A est négatif

Soit "a" réduit dans l'interval-

le $(-\frac{1}{2} \text{Log}_{10} \frac{1}{2} \text{Log}_{10})$

Calcul de l'approximation rationnelle donnant e^a.

Le dénominateur et le numérateur sont calculés suivant le schéma de

Horner à l'aide de la même séquence. Avant chaque exécution, il y a

l'enregistrement de l'instruction à exécuter à la sortie de cette sé-
quence.

0063 69 0016 0020

(0016) = 21 0022 0025

Stockage du dénominateur après le pre-
mier passage.

(4) 0020 24 0024 0028

(0024) = Instruction à exécuter à la

sortie de la boucle (4).

Changement de signe de la variable.

0028 61 0060 0015

0015 21 0060 0064

(0060) = "a" cadré 1 au premier passage

= "a" cadré 1 au second passage

(0017) = "K₆" cadré 0.

0064 19 0017 0038

0038 35 0001 0045

0045 10 0048 0053

(0048) = "K₅" cadré 0.

0053 60 8003 0061

0061 19 0060 0031

0031 35 0001 0039

0039 10 0042 0001

(0042) = "K₄" cadré 0.

0001 60 8003 0009

0009 19 0060 0032

0032 35 0001 0043

0043 10 0046 0051

(0046) = "K₃" cadré 0.

0051 60 8003 0059

0059 19 0060 0034

0034 35 0001 0044

0044 10 0002 0007

0007 60 8003 0018

0018 19 0060 0035

0035 35 0001 0052

0052 10 0055 0012

0012 60 8003 0021

0021 19 0060 0036

0036 10 0014 0024

(0002) = "K₂" cadré 0.

(0055) = "K₁" cadré 0.

(0014) = 10 0000 0000

Sortie de (4) :

(0024) = 21 0022 0025

(0022) = $\sum_{n=0}^6 K_n (-a)^n$ cadré 1.

Préparation du second passage.

(0030) = 30 0001 0004

Cadrage 2 du numérateur en vue de la
division pour obtenir e^a.

Renvoi à (4) pour :

- l'enregistrement de cette instruction

- le changement de signe de la variable

- le calcul du numérateur.

Sortie de (4) :

(0034) = 30 0001 0004

(8000) = $\sum_{n=0}^6 K_n (a)^n$ cadré 2.

(8002) = e^a cadré 1.

(0062) = 10 $\frac{1}{2}$

Cadrage 11 dans l'accumulateur.

(0037) = Puissance de 10 du résultat.

(0003) = Instruction de sortie.

Mémoires de travail

à contenus constants

Constantes intervenant dans la réduction d'intervalle.

(0012) = 50 0000 0000
(0023) = 25 0000 0000
(0002) = 12 5000 0000
(0016) = 23 0258 5093 ($\log_{10}$)
(0024) = 43 4294 4819 ($\log_{10} e$)
(0030) = 00 0000 0001
(0031) = 00 0002 0000 (Mot de contrôle pour $\xi_3 = 1$)
(0060) = 00 0001 0000 (Mot de contrôle pour $\xi_3 = 1$)
(0076) = 19 0050 0062 (Instruction qui sera modifiée suivant la

valeur de α)

Coefficients de l'approximation polynomiale de $e^3 \log_{10}$

(0014) = 10 0000 0000 (K_0)
(0033) = 23 0258 5121 (K_1)
(0074) = 26 5094 9074 (K_2)
(0072) = 20 3486 4047 (K_3)
(0070) = 11 7124 9183 (K_4)
(0065) = 05 4124 4748 (K_5)
(0008) = 02 0760 8277 (K_6)

Table de 10^{α}

(0050) = 13 3352 1432 ($10^{\frac{1}{3}}$)
(0051) = 23 7137 3706 ($10^{\frac{2}{3}}$)
(0052) = 42 1696 5034 ($10^{\frac{1}{2}}$)
(0053) = 74 9894 2093 ($10^{\frac{2}{5}}$)

à contenus variables

(0003) = Instruction de sortie du sous-programme.

(0018) = Mot de contrôle dans la réduction (i=3)

Son contenu est :

00 0000 0000 si $\xi_3 = -1$.

00 0001 0000 si $\xi_3 = 1$

(0026) = f_3 cadré 0.

(0039) = Puissance de 10 du résultat, cadrée 10. (" m " peut avoir 10 chiffres significatifs).

Notation

A : Argument cadré 11 dans l'accumulateur à compter de la position gauche. A peut avoir jusqu'à 19 chiffres significatifs.

$a_1 = f_1 \cdot \log_{10}$: Argument réduit dans l'intervalle $(-\frac{1}{2}, \frac{1}{2}) \log_{10}$, $\frac{1}{2} \log_{10}$.

Suivant son signe A est mis sous la forme :

$A = (N+F) \cdot \log_{10}$ si A est positif.

$A = [-(N+1) + (1-F)] \cdot \log_{10}$ si A est négatif.

avec "N" entier positif et "F" nombre fractionnaire positif. ($0 \leq F < 1$)

L'exponentielle devient : $e^A = 10^{m \cdot f \cdot \log_{10}}$

avec " m " = N et $f_0 = F$ si A est positif.

avec " m " = -(N+1) et $f_0 = 1-F$ si A est négatif.

Réduction de " f_0 " dans l'intervalle $(-\frac{1}{2}, \frac{1}{2})$

Nous choisissons i = 3.

-réduction (i=1)

$$f_1 = f_0 - \frac{1}{2} \quad |f_1| \leq \frac{1}{2} \quad e^{f_0 \log_{10}} = 10^{\frac{1}{2}} \cdot e^{f_1 \log_{10}}$$

-réduction (i=2)

$$\begin{aligned} f_2 &= f_1 - \frac{1}{4} & |f_2| &\leq \frac{1}{4} & e^{f_0 \log_{10}} &= 10^{\frac{1}{4}} \cdot e^{\frac{f_1}{2} \log_{10}} \\ f_1 &= f_1 + \frac{1}{4} \end{aligned}$$

$\xi_2 = 1$ si f_1 positif ou nul; $\xi_2 = -1$ si f_1 négatif.

-réduction (i=3)

$$\begin{aligned} f_3 &= f_2 - \frac{1}{8} & |f_3| &\leq \frac{1}{8} & e^{f_0 \log_{10}} &= 10^{\frac{1}{8}} \cdot e^{\frac{f_2}{4} \log_{10}} \\ f_2 &= f_2 + \frac{1}{8} \end{aligned}$$

$\xi_3 = 1$ si f_2 positif ou nul; $\xi_3 = -1$ si f_2 négatif.

La forme finale est :

$$e^A = 10^m \cdot 10^{\alpha} \cdot e^{f_3 \log_{10}} \quad \text{avec } \alpha = +\frac{1}{3} + \frac{\xi_2}{4} + \frac{\xi_3}{8}$$

En fin de calcul, $10^{\alpha} \cdot e^{f_3 \log_{10}}$ est cadré 11 dans l'accumulateur.

La puissance de 10 de ce résultat est chargée en 8001, avec son signe.

(Le chiffre unité est en position 1 du distributeur).

(0040) = Mot de contrôle dans la réduction (i=2)

Son contenu est :

00 0000 0000 si $\xi_k = -1$.

00 0002 0000 si $\xi_k = 1$.

= I9 005k 0062

Instruction modifiée suivant la valeur de ϵ . "k" peut

prendre les valeurs 0,1,2,3 et la mémoire adressée (005k) contient 10^{ϵ} , avec $\epsilon = \frac{2k+1}{8}$ (voir table de 10^{ϵ})

Instructions

0000 24 0003 0006 (0003) = Instruction de sortie.

0006 45 0010 0011 (1)

L'argument est nul. Enregistrement de 1 en 8002 et de 0 en 8001.

0011 I5 0014 0019 (0014) = I0 0000 0000

0019 69 8003 0003 Sortie pour un argument nul.

Calcul de " $f_0 \cdot \log_{10}$ " et de la puissance de 10 du résultat (stockée en 0039).

(1) 0010 46 0013 0064 (2) Test du signe de A pour choisir la forme

0013 I4 0016 0027 (0016) = $\log_{10}$

0027 I6 0030 0035 (0030) = 00 0000 0001

0035 20 0039 0042 (0039) = $-(N+1)$ cadré 10. ($A < 0$)

0042 60 8003 0049 (8003) = $-F \cdot \log_{10}$

0049 I0 0018 0021 (3) (8003) = $f_0 \cdot \log_{10}$ cadré 1.

(2) 0064 I4 0016 0077

0077 20 0039 0043 (0039) = N cadré 10. ($A > 0$)

0043 60 8003 0021 (8003) = $f_0 \cdot \log_{10}$ cadré 1.

Isolément de " f_0 " et réduction dans l'intervalle $(-\frac{1}{8}, \frac{1}{8})$, en trois étapes. A la fin des réductions partielles (i=2) et (i=3), il y a enregistrement d'un mot de contrôle indiquant le signe de " ϵ_1 ".

(3) 0021 I9 0024 0045 (0024) = $\log_{10}$ cadré 0.

0045 35 0001 0001

0001 60 8003 0009 (8003) = f_0 cadré 0.

0009 II 0012 0017 (0012) = 50 0000 0000

Réduction (i=1). 8003 contient " f_4 "

0017 46 0020 0071 (4) Réduction (i=2).

0020 I0 0023 0028 (0023) = 25 0000 0000

8003 contient " f_2 ".

0028 69 8002 0036 (8002) = 00 0000 0000

Mot de contrôle correspondant à $\epsilon_2 = -1$.

(4) 0071 II 0023 0078 (8003) = f_2

0078 69 0031 0036 (0031) = 00 0002 0000

Mot de contrôle correspondant à $\epsilon_2 = 1$.

(0040) = Mot de contrôle du signe de ϵ_2 .

0044 46 0047 0048 (6) Réduction (i=3)

0047 I0 0002 0007 (0002) = I2 5000 0000

8003 contient " f_3 ".

0007 69 8002 0015 (7) (8002) = 00 0000 0000

Mot de contrôle correspondant à $\epsilon_3 = -1$.

(6) 0048 II 0002 0057 (8003) = f_3

0057 69 0060 0015 (0060) = 00 0001 0000

Mot de contrôle correspondant à $\epsilon_3 = 1$.

(7) 0015 24 0018 0022 (0018) = Mot de contrôle du signe de ϵ_3 .

0022 21 0026 0029 (0026) = f_3 cadré 0.

Détermination de 10^{ϵ} , à l'aide des mots de contrôle.

Suivent la réduction, " ϵ " peut prendre quatre valeurs différentes. Chacune d'elles est référée par le résultat de la somme des contenus des mémoires (0040) et (0018):

Valeur de ϵ : $\frac{1}{8} = \frac{1}{2} - \frac{1}{4} - \frac{1}{8}$. Somme des mots de contrôle: 00 0000 0000

" $\frac{3}{8} = \frac{1}{2} - \frac{1}{4} + \frac{1}{8}$ " " : 00 0001 0000

" $\frac{5}{8} = \frac{1}{2} + \frac{1}{4} - \frac{1}{8}$ " " : 00 0002 0000

" $\frac{7}{8} = \frac{1}{2} + \frac{1}{4} + \frac{1}{8}$ " " : 00 0003 0000

Cette somme additionnée à l'instruction : I9 0050 0062, permet par modification de l'adresse facteur, de choisir dans une table précalculée la valeur 10^{ϵ} convenable.

0029 60 0040 0046 (0046) = 00 0000 0000 ($\epsilon_2 = -1$)
 = 00 0002 0000 ($\epsilon_2 = 1$)
 0046 10 0018 0073 (0018) = 00 0000 0000 ($\epsilon_3 = -1$)
 = 00 0001 0000 ($\epsilon_3 = 1$)
 (8003) = Modification à apporter à
 l'adresse facteur.
 0073 10 0076 0032 (0076) = 19 0050 0062
 Instruction à modifier.
 0032 21 0040 0004 (0040) = 19 005(k) 0062 (k=0,1,2,3)
 Calcul de $e^3 \log_e 10 = \sum_{n=0}^6 K_n \cdot r^n$ suivant le schéma de Horner. En
 fin de calcul il y a exécution de l'instruction modifiée et chargement
 de la puissance de 10 du résultat dans le distributeur.
 0004 60 0008 0063 (0008) = K_6 cadré 1.
 0063 19 0026 0054 (0026) = r_3 cadré 0;
 0054 60 8003 0061 (0065) = K_5 cadré 1.
 0061 10 0065 0069
 0069 19 0026 0005
 0005 60 8003 0066
 0066 10 0070 0025 (0070) = K_4 cadré 1.
 0025 19 0026 0055
 0055 60 8003 0067
 0067 10 0072 0079 (0072) = K_3 cadré 1.
 0079 19 0026 0056
 0056 60 8003 0068
 0068 10 0074 0080 (0074) = K_2 cadré 1.
 0080 19 0026 0058
 0058 60 8003 0075
 0075 10 0033 0037 (0033) = K_1 cadré 1.
 0037 19 0026 0059
 0059 60 8003 0034
 0034 10 0014 0040 (0014) = K_0 cadré 1.
 (8003) = $e^3 \log_e 10$ cadré 1.
 (0040) = Instruction modifiée.

(0040) 19 005k 0062 (8000) = $10^{\alpha} \cdot e^3 \log_e 10$ cadré 2.
 0062 35 0001 0039
 0039 65 8003 0041 Cadrage 11 de $10^{\alpha} \cdot e^3 \log_e 10$ dans
 l'accumulateur.
 0041 69 0039 0003 (0039) = Puissance de 10 du résultat
 cadrée 10.
 (0003) = Instruction de sortie.

Notation

A : Argument cadré 11 dans l'accumulateur à compter de la position gauche. A peut avoir jusqu'à 19 chiffres significatifs.

a : Argument réduit dans l'intervalle $(-\frac{\pi}{2}, \frac{\pi}{2})$.

Le calcul de Cosinus est ramené au calcul de Sin. $(A, \frac{\pi}{2})$.

Détermination de "a".

-réduction dans l'intervalle $(-2\pi, 2\pi)$

elle est obtenue par division de A ou $A + \frac{\pi}{2}$ par 2π . A représente le reste; le quotient est ignoré: $|A_2| < 2\pi$ $\sin(A) = \sin(A_2)$

-réduction dans l'intervalle $(-\pi, \pi)$

$$A_2 \geq 0 \quad A_1 = A_2 - \pi$$

$$A_2 < 0 \quad A_1 = A_2 + \pi$$

$$|A_1| \leq \pi \quad \sin(A) = -\sin(A_1)$$

-réduction dans l'intervalle $(-\frac{\pi}{2}, \frac{\pi}{2})$

$$A_1 \geq 0 \quad a = A_1 - \frac{\pi}{2}$$

$$A_1 < 0 \quad a = A_1 + \frac{\pi}{2}$$

$$|a| \leq \frac{\pi}{2} \quad \sin(A) = \pm \sin(a)$$

avec "ε" = -1 si A₁ est positif ou nul ; "ε" = 1 si A₁ est négatif.

Changement de variable

$$a = \frac{\pi}{2} X \quad \text{avec } |X| \leq 1.$$

Cos(a) est obtenu par $\sum_{n=0}^{\infty} K_n X^{2n}$ et cadré 11 dans l'accumulateur avec le signe convenable.

La première instruction du sous-programme se trouve en 0000 ou 0001 suivant qu'il s'agit du calcul de sinus ou de cosinus.

Mémoires de travailà contenus constants

Constantes pour la détermination de "a".

$$(0009) = 62 \ 8318 \ 5307 \ (2\pi)$$

$$(0010) = 15 \ 7079 \ 6327 \ (\frac{\pi}{2})$$

$$(0034) = 31 \ 4159 \ 2654 \ (\pi)$$

Coefficients de l'approximation polynomiale donnant cos(a)

$$(0044) = 10 \ 0000 \ 0000 \ (K_0)$$

$$(0026) = -12 \ 5370 \ 0534 \ (K_2)$$

$$(0002) = 25 \ 3669 \ 3225 \ (K_4)$$

$$(0051) = -02 \ 0862 \ 6875 \ (K_6)$$

$$(0050) = 00 \ 0917 \ 7239 \ (K_8)$$

$$(0041) = -00 \ 0023 \ 8245 \ (K_{10})$$

à contenus variables

(0003) = Instruction de sortie du sous-programme.

(0024) = Mot d'orientation

son contenu est :

Instruction de sortie si "ε" = 1.

66 8002 0003 si "ε" = -1.

(0025) = X² cadré 0.

Instructions

0000 24 0003 0006 (1) Instruction de départ pour le calcul de sinus.

0001 24 0003 0007 Instruction de départ pour le calcul de cosinus.

0007 15 0010 0006 (0010) = $\frac{\pi}{2}$

Nous sommes ramenés au calcul de sinus.

Réduction de l'intervalle.

(0009) = 2π

(8003) = A₂ cadré 1.

Test du signe de A₂

(0034) = π

(8003) = A₁ cadré 1.

Test du signe de A₁

(0010) = $\frac{\pi}{2}$

(0018) = 66 8002 0003

Changement de signe de cos.a (ε = -1)

(4) 0042 10 0010 0016

0016 69 0003 0021

(0003) = Instruction de sortie

Le résultat a même signe que cos(a) (ε = 1)

(0024) = Mot d'orientation : sortie ou

changement de signe préalable.

(5) 0021 24 0024 0028

Changement de variable et résolution des cas où l'argument est un arc remarquable.

0028 45 0032 0033 (6) L'argument est nul. Le résultat correspondant est : $\cos(k\pi)$ ou $\sin[(2k+1)\frac{\pi}{2}]$. Il y a renvoi à l'addition de "1" dans le calcul de $\cos(a)$. L'argument est différent de zéro. Calcul de la variable X .

0032 30 0001 0040
0040 64 0010 0022
0022 35 0001 0029
0029 44 0035 0036 (7)

La partie entière de X est non nul.
 $|X| = 1$. Le résultat correspondant est : $\cos[(2k+1)\frac{\pi}{2}]$ ou $\sin(k\pi)$. Enregistrement de zéros dans l'accumulateur et sortie.
(0003) = Instruction de sortie.

0035 65 8002 0003
Calcul de $\cos(a) = \cos\left(\sum_{n=0}^5 K_n X^n\right)$ suivant le schéma de Horner.

(7) 0036 60 8002 0045 (8002) = X cadré 0.

0045 19 8003 0020

0020 21 0025 0037 (0025) = X^2 cadré 0.

0037 60 0041 0046 (0041) = K_{10} cadré 0.

0046 19 0025 0047

0047 10 0050 0005 (0050) = K_8 cadré 0.

0005 60 8003 0013

0013 19 0025 0048

0048 10 0051 0003 (0051) = K_6 cadré 0.

0008 60 8003 0017

0017 19 0025 0049

0049 10 0002 0011 (0002) = K_4 cadré 0.

0011 60 8003 0023

0023 19 0025 0052

0052 30 0001 0012 Cadrage 1 pour l'addition de K_2 .

0012 10 0026 0038 (0026) = K_2 cadré 1.

0038 60 8003 0004
0004 19 0025 0014
0014 65 8003 0033 Cadrage 11 dans l'accumulateur.
(6) 0033 15 0044 0024 (0044) = K_0 cadré 1.
(8002) = $\cos(a)$ cadré 1.
(0024) = Mot d'orientation: 66 8002 0003
si $W_{E''} = -1$.
0018 66 8002 0003 Changement du signe de $\cos(a)$
(0024) = (0003) si $W_{E''} = 1$.
(0003) = Instruction de sortie.

Notation

A : Argument cadré 11 dans l'accumulateur à compter de la position gauche et limité à 10 chiffres significatifs.

a : Argument réduit dans l'intervalle (0, 1), bornes exclues.

Détermination de "a"

$$- \quad A = \varepsilon |A| \quad \text{Arc tg } A = \varepsilon \text{Arc tg } |A|$$

avec "ε" = 1 si A est positif ; "ε" = -1 si A est négatif.

- |A| est comparé à l'unité :

$$\begin{aligned} |A| < 1 & \quad a = |A| & \quad \text{Arc tg } A = \varepsilon \text{Arc tg } a \\ |A| > 1 & \quad a = \frac{1}{|A|} & \quad \text{Arc tg } A = \varepsilon \left(\frac{\pi}{2} - \text{Arc tg } \frac{1}{|A|} \right) \\ |A| = 1 & \quad a = 1 & \quad \text{Arc tg } A = \varepsilon \frac{\pi}{4} \end{aligned}$$

Changement de variable

$$t = \frac{a - a_0}{1 + a \cdot a_0} \quad \text{Arc tg } a = \text{Arc tg } a_0 + \text{Arc tg } t$$

avec : a = 0,xyz...

$$a_0 = 0, x50000000$$

Arc tg a₀ est donné par une table précalculée de 10 valeurs.

Arc tg t est donné par un développement polynomial de trois termes.

Le résultat final est cadré 11 dans l'accumulateur.

La première instruction du sous-programme se trouve en 0000.

Mémoires de travail

à contenus constants

$$(0016) = 07\ 8539\ 8163\ \left(\frac{\pi}{4}\right)$$

$$(003E) = 00\ 0000\ 0009$$

$$(0038) = 50\ 0000\ 0000$$

$$(0049) = 15\ 7079\ 6327\ \left(\frac{\pi}{2}\right)$$

(0075) = 10 0049 5009 quantités qui, additionnées à "a₀", donne l'ins-truction qui permet l'addition de Arc tg a₀ convenable.

Coefficients de l'approximation de Arc tg t.

$$(0065) = 10\ 0000\ 0000\ (K_4)$$

$$(0068) = 33\ 3333\ 3333\ (K_3)$$

$$(0081) = 20\ 0000\ 0000\ (K_2)$$

Table de Arc tg a₀

$$(0050) = 00\ 4995\ 8396\ (\text{Arc tg } 0,05 \text{ cadré } 1)$$

$$(0051) = 01\ 4888\ 9948\ (\quad " \quad 0,15 \quad " \quad)$$

$$(0052) = 02\ 4497\ 8665\ (\quad " \quad 0,25 \quad " \quad)$$

$$(0053) = 03\ 3667\ 4819\ (\quad " \quad 0,35 \quad " \quad)$$

$$(0054) = 04\ 2285\ 3926\ (\quad " \quad 0,45 \quad " \quad)$$

$$(0055) = 05\ 0284\ 3211\ (\quad " \quad 0,55 \quad " \quad)$$

$$(0056) = 05\ 7637\ 5221\ (\quad " \quad 0,65 \quad " \quad)$$

$$(0057) = 06\ 4350\ 1109\ (\quad " \quad 0,75 \quad " \quad)$$

$$(0058) = 07\ 0449\ 4064\ (\quad " \quad 0,85 \quad " \quad)$$

$$(0059) = 07\ 5976\ 2755\ (\quad " \quad 0,95 \quad " \quad)$$

à contenus variables

$$(0007) = \text{Instruction de sortie du sous-programme.}$$

$$(0008) = a_0 \text{ cadré } 1.$$

$$= t^3 \text{ cadré } 0.$$

$$(0009) = 30\ 000(9-n)\ 0062$$

Instruction utilisée pour cadrer 11 l'inverse de |A|.

= Mot d'orientation

Son contenu est :

Instruction contenue en 0030 si |A| est inférieur à 1

61 8003 0042 si |A| supérieur à 1.

(0017) = Partie de A contenue en 8002. (Changement du signe de A)

= |A| cadré 1 dans le calcul de l'inverse.

= a cadré 0.

= t cadré 0.

(0030) = Mot d'orientation

Son contenu est :

66 8003 0007 si "ε" = -1.

65 8003 0007 si "ε" = 1.

(0074) = 1+a.a₀ dans le calcul de "t".

Instructions

0000	45 0004 8001	Sortie pour un argument nul.	0005	22 0009 0012	(0009) = 30 000(9-n) 0002
0004	24 0007 0010	(0007) = Instruction de sortie.			Instruction de décalages permettant le
0010	46 0013 0014 (1)	Test du signe de A.			cadre 11 de l'inverse de A
0013	20 0017 0020	A négatif. Changement de son signe.			(Le décalage n'est pas avec arrondi à
0020	61 8003 0027				cause du cas fréquent où 9-n=0)
0027	16 0017 0021		0012	60 0065 0019	(0065) = 10 0000 0000
0021	69 0024 0077 (2)	(0024) = 66 8003 0007	0019	30 0001 0025	(8002) = $\frac{4}{M}$ cadré 1.
		Changement de signe et cadre 11 du	0025	64 0017 0009	Cadrage 11 de $\frac{4}{ A }$
		résultat. ($\xi = -1$)	(0009)	30 000(9-n) 0002	
(1)	0014 69 0067 0077	(0067) = 65 8003 0007	0002	35 0001 0060	
		A positif. Seulement cadre 11 du résul-	0060	44 0063 0064 (4)	
		tat. ($\xi = 1$).			L'Arc est égal à $\frac{\pi}{4}$ en valeur absolue.
(2)	0077 24 0030 0033	(0030) = Mot d'orientation. (Distinc-			Enregistrement de cette valeur en 8003.
		tion des cas $\xi = -1$ et $\xi = 1$)			Renvoi à 0030 pour la détermination du
		Comparaison de A avec l'unité. Il suffit	0063	60 0016 0030	signe.
		de tester la nullité de la partie en-			(0016) = $\frac{\pi}{4}$
		tière. A est alors cadré 10 dans 8000.	(4)	0064 69 0018 0044	(0018) = 61 8003 0042
0033	35 0001 0039				Changement du signe de Arc $\text{tg } \frac{4}{ A }$ pour
0039	44 0043 0044 (3)				former $\frac{\pi}{2} - \text{Arc } \text{tg } \frac{4}{ A }$ ($ A > 1$)
		A est inférieur à 1. ("a" = A). Renvoi	(3)	0044 24 0009 0062	(0009) = Mot d'orientation. (Distinction
		à la fin de la réduction dans l'inter-			des cas $ A < 1$ et $ A > 1$)
		valle (0 1). Le contenu de (0030) se-			(0017) = a cadré 0.
		ra chargé dans (0009).	0062	20 0017 0071	
		A est supérieur ou égal à 1. Calcul de l'inverse de A .			Détermination de "a". Etablissement de l'instruction permettant le
		A cadré 10 est mis sous la forme : $ A = M \cdot 10^{-n}$ avec $1 \leq M < 10$. (n = nom-			choix convenable de la valeur de Arc $\text{tg } a_0$ dans la table précalculée.
		bre de décalages à gauche pour obtenir M cadré 1 en 8003). (9-n) déca-			"a" cadré 5 est additionné à la quantité 10 0049 5009 :
		lages à droite de l'inverse de M cadré 1 en 8002 donnent l'inverse de			00 000x 5000
		A cadré 11 en 8000.			10 0049 5009
0043	36 0000 0015				
0015	21 0017 0070	(0017) = M cadré 1.			La somme : 10 005x 0009 donne l'instruction cherchée. La
0070	66 8002 0029	(8002) = 00 0000 000n			mémoire adressée 005(x) contient la valeur de Arc $\text{tg}(0, x5)$.
0029	15 0032 0037	(0032) = 00 0000 0009	0071	35 0001 0028	Isolément de la première décimale de "a".
0037	35 0004 0047		0028	60 8003 0035	(8003) = 00 0000 000x
			0035	15 0038 0045	(0038) = 50 0000 0000
			0045	30 0002 0003	
0047	69 0001 0005	(0001) = 30 0000 0002	0003	20 0008 0011	(0008) = a_0 cadré 1.

(0009) = (0030) si $|A|$ inférieur à 1.
 (0030) = Mot d'orientation : 66 8003 0007
 ($\varepsilon = -1$)
 Changement de signe de Arc tg A et ca-
 drage ll du résultat.
 (0030) = Mot d'orientation : 65 8003 0007
 Cadrage ll du résultat.
 (0007) = Instruction de sortie.

0024 66 8003 0007

0067 65 8003 0007

Cadrage B de "a".
 (0075) = IO 0049 5009
 (0083) = IO 005(x) 0009

Addition de Arc tg a₀.

Calcul de "t", de son carré et de Arc tg t suivant le schéma de Horner

(0017) = a cadré 0.
 (0008) = a₀ cadré 1.
 (0065) = IO 0000 0000
 (0074) = l + a₀ cadré 1.

(8002) = t cadré 0.

(0008) = t² cadré 0.

(0081) = K₅ cadré 0.
 (0068) = K₃ cadré 0.

(0065) = K₁

(8003) = Arc tg t cadré 1.

Addition de Arc tg a₀ à Arc tg t.

(8003) = Arc tg a

(0009) = Mot d'orientation : 61 8003 0042

(|A| supérieur à 1)

(0049) = $\frac{\pi}{2}$ (8003) = $\frac{\pi}{2}$ - Arc tg $\frac{1}{|A|}$. Renvoi à 0030

0011 30 0004 0022
 0022 15 0075 0079
 0079 20 0083 0036

0036 60 0017 0072
 0072 19 0008 0080

0080 10 0065 0069

0069 21 0074 0078

0078 60 0017 0023

0023 30 0001 0031

0031 11 0008 0086

0066 64 0074 0085

0085 60 8002 0046

0046 24 0017 0073

0073 19 8001 0048

0048 21 0008 0061

0061 60 8003 0026

0026 19 0081 0006

0006 11 0068 0076

0076 60 8003 0034

0034 19 0008 0082

0082 30 0001 0040

0040 10 0065 0084

0084 60 8003 0041

0041 19 0017 0083

(0083) IO 005(x) 0009

0018 61 8003 0042

0042 IO 0049 0030

Sous-programme SFVF

Ce sous-programme inclut dans chacun des sous-programmes fonctionnels (simple précision) permet leur utilisation en virgule flottante du type FLAIR.

SFVF transforme l'argument donné en virgule flottante en un argument écrit en virgule fixe et cadré 11 dans l'accumulateur à compter de la position 20, assure l'entrée dans le sous-programme établi en virgule fixe puis reprend le résultat obtenu pour le transformer en écriture "virgule flottante".

SFVF étant commun à toutes les fonctions, n'est pas translatable comme l'est chaque sous-programme fonctionnel qui assure son chargement. Sa zone d'enregistrement est toujours (I735-I799). Il convient donc d'indiquer dans une mémoire d'orientation le facteur de translation "Δ" utilisé pour le chargement du sous-programme fonctionnel correspondant.

Pour entrer dans SFVF, 800I, 8002, 8003 et la mémoire adressée (I778) doivent contenir les données suivantes :

```

800I = Instruction de sortie.
8002 = 00 0000 0000
8003 = 44 xxxx xxxx (Argument)
I778 = +69 I757 Δ si la fonction à calculer est la "Racine Carrée" (RC) ou l'"Exponentielle" (EX).
      = -69 I757 Δ si la fonction à calculer est le "Logarithme népérien" (LE) ou le "Sinus" (SN) ou l'"Arc Tangente" (AT) ou l'"Arc Sinus" (AS).
      = -69 I757 Δ+1 si la fonction à calculer est le "Cosinus" (CN).
  
```

Pour toutes les fonctions, la première instruction se trouve en I750.

Mémoires de travail

à contenus constants

```

(I736) = 00 0000 006I
(I747) = 00 0000 0020
(I76I) = 00 0000 00I0
(I782) = 5I 0000 0000
(I796) = 6I 0000 0000
  
```

à contenus variables

(I753) = Instruction de sortie du sous-programme.

(I770) = Instruction de décalages utilisée pour cadrer il l'argument obtenu en virgule fixe.

Son contenu est :

```

3I 0000 I778. si 44 = 4I
3I 00(44-5I) I778 si 4I < 44 < 5I
35 00(44-5I) I778 si 5I ≤ 44 < 6I
  
```

= Puissance de 10 du résultat ou nombre de positions décalées. Cette quantité sert à la détermination de l'exposant du résultat final.

(I773) = Mantisse de l'argument cadrée extrême gauche. (xx xxxx xx00)

Si 44 est inférieur à 4I, la mantisse est ignorée et remplacée par 00 0000 0000.

= Mantisse du résultat (00 yyyy yyyy).

(I778) = Mot d'orientation.

Son contenu est :

```

+69 I757 Δ pour RC, EX.
-69 I757 Δ pour LE, SN, AT, AS.
-69 I757 Δ+1 pour CN.
  
```

(I788) = 5I 0000 0000 pour RC, EX et les autres fonctions si le résultat est inférieur à 10.

6I 0000 0000 pour les autres fonctions si le résultat est supérieur ou égal à 10.

Cette donnée permet la détermination de l'exposant du résultat final.

(I797) = Mot d'orientation.

Avant le calcul de la fonction en virgule fixe son contenu est :

44 I743 I746 (LX, SN, AT, AS, CN)

24 I770 I779 (RC, EX)

Après le calcul de la fonction :

68 8002 I795 (LX, SN, AT, AS, CN)

15 I788 I798 (RC, EX)

Instructions

I750 24 I753 I756 (I753) = Instruction de sortie.

I756 65 8003 I763

I763 35 0002 I769 Isolement de l'exposant $\alpha\alpha$.

I769 20 I773 I776 (I773) = Mantisse cadrée extrême gauche

I776 67 8003 I783

I783 16 I736 I741 (I736) = 00 0000 0061

Comparaison de l'exposant $\alpha\alpha$ avec 61.

I741 46 I744 I745
(1)

L'exposant est supérieur ou égal à 61. L'argument a donc plus de 10 chiffres significatifs avant la virgule : sous peine de perdre des chiffres en tête, la transformation en virgule fixe est impossible en simple précision. Il y a arrêt programmé.

(I3) I745 01 I745 9999 Le même arrêt sera utilisé dans le cas où l'exposant du résultat serait en dehors des limites 00 et 99 (voir en I3).

Comparaison de l'exposant $\alpha\alpha$ avec 41.

(1) I744 15 I747 I751 (I747) = 00 0000 0020

I751 46 I754 I755 (2)

L'exposant est inférieur à 41. L'argument écrit en virgule fixe se présente avec au moins 10 zéros en tête : il est donc considéré comme nul en simple précision.

I754 21 I773 I777 (3) (I773) = 00 0000 0000

L'exposant est supérieur ou égal à 41.

(2) I755 45 I758 I759
(4)

L'exposant égale 41. L'argument en virgule fixe est obtenu cadré 11 par un décalage à droite de 10 positions de la mantisse. L'arrondi permet de faire apparaître un chiffre significatif en position unitaire dans le cas où le premier chiffre de la mantisse est supérieur ou égal à 5.

I759 69 I762 I766 (5) (I762) = 31 0000 I778

L'exposant est compris entre 41 et 61. Le nombre de positions dont il faut décaler la mantisse pour obtenir l'argument en virgule fixe cadré 11 est de : $\alpha\alpha - 51$. Suivant le signe de cette quantité, le décalage aura lieu vers la droite ou vers la gauche.

(4) I758 16 I761 I765 (I761) = 00 0000 0010

(8002) = 00 0000 00($\alpha\alpha - 51$)

I765 35 0004 I775

I775 46 I759 I777 (3) Test du signe de ($\alpha\alpha - 51$) pour déterminer le sens du décalage.

I759 69 I762 I766 (5) (I762) = 31 0000 I778

(3) I777 69 I780 I766 (I780) = 35 0000 I778

(5) I766 22 I770 I774 (I770) = 31 0000 I778 si $\alpha\alpha = 41$

31 00($\alpha\alpha - 51$) I778 41 < $\alpha\alpha$ < 51

35 00($\alpha\alpha - 51$) I778 51 < $\alpha\alpha$ < 61

(I778) = Mot d'orientation

I774 60 I778 I784

+69 I757 Δ (RC, EX)

-69 I757 Δ (LX, SN, AT, AS)

-69 I757 $\Delta + 1$ (CN)

Signe "+": le résultat en virgule fixe est obtenu en 8000 avec sa puissance de 10 en 8001.

Signe "-": le résultat en virgule fixe est obtenu uniquement en 8000. Le test du signe du mot d'orientation permet d'enregistrer l'instruction à exécuter après l'obtention du résultat suivant que le contenu de 8001 concerne ou non le résultat.

I784 46 I737 I738 (8)

I737 69 I740 I794 (9) (I740) = 44 I743 I746

(8) I738 69 I791 I794 (I791) = 24 I770 I779

(9) I794 24 I797 I752 (I797) = Mot d'orientation.

Préparation de l'entrée dans le sous-programme fonctionnel correspondant pour le calcul de la fonction en virgule fixe.

I752	65	I773	I770	(I773) = Mantisse de l'argument. (remplacée par 000000000 si l'exposant est inférieur à 41)
(I770)	31 35	xxxx	I778	Décalage de la mantisse pour obtenir l'argument cadré 11 dans l'accumulateur. (I757) = 45 I797 I753
(I778)	69	I757	Δ ou $\Delta+1$	Instruction qui sera exécutée après l'obtention du résultat en virgule fixe.
$\Delta + 0000$ ou $\Delta + 0001$ est l'adresse de la première instruction du sous-programme fonctionnel en virgule fixe.				
$\Delta (\Delta+1)$ Sous-programme fonctionnel virgule fixe. (I757)				
I757	45	I797	I753	Le résultat en virgule fixe sort cadré 11 dans l'accumulateur.
Sortie si le résultat est nul. (pas de transformation d'écritures)				
(I797) = Mot d'orientation : 24 I770 I779 (RC, KK)				
I791	24	I770	I779	(I770) = Puissance de 10 du résultat.
L'exposant $\beta\beta$ du résultat en virgule flottante est égal à :				
51+ Puissance de 10.				
I779	69	I782	I735	(I782) = 51 0000 0000
I735	24	I788	I742	(I795) = 15 I788 I798
I742	69	I795	I748 (10)	Instruction utilisée pour former $\beta\beta$. Elle sera également chargée en I797.
(I797) = Mot d'orientation :				
44 I743 I746 (LE, SN, AT, AS, CN)				
Comparaison du résultat avec 10.				
I740	44	I743	I746 (11)	

Le résultat est inférieur à 10. Il est transféré en 8003. L'exposant $\beta\beta$ est égal à 51 - |n|. ("n" est le nombre de décalages à effectuer pour éliminer les zéros en tête).

I746 60 8002 I760

I760 69 I782 I785 (12) (I782) = 51 0000 0000

Le résultat est supérieur ou égal à 10. (Seul cas possible: calcul de LE)

L'exposant $\beta\beta$ est égal à 61 - |n|.

(11) I743 69 I796 I785 (I796) = 61 0000 0000

(12) I785 24 I788 I792

I792 36 0000 I767 Elimination des zéros en tête.

I767 20 I770 I781 (I770) = 00 0000 000n

I781 65 8003 I739 (8002) = Résultat normalisé.

I739 69 I793 I743 (I793) = 68 8002 I795

Instruction donnant -|n|

Cadrage de la mantisse et calcul de l'exposant du résultat final.

(10) I743 24 I797 I764 (I797) = Mot d'orientation

I764 31 0002 I786 Le résultat normalisé est tronqué à 8 chiffres avec arrondi du dernier.

I786 20 I773 I787 (I773) = 00 YYY YYY

I787 80 I770 I789

I789 30 0002 I797

(I797) = Mot d'orientation :

68 8002 I795 (LE, SN, CN, AT, AS)

I793 68 8002 I795 (I797) = (I795) si RC, KK.

I795 15 I788 I798 (I788) = 51 0000 0000 ou 61 0000 0000

(8000) = Exposant $\beta\beta$ cadré 12.

Vérification de la valeur de l'exposant par rapport aux limites 00 et 99.

I798 46 I745 I768 (13) Exposant négatif. Renvoi à l'arrêt programmé. (Voir début du programme)

I768 44 I745 I772 (13) Exposant supérieur à 99. Renvoi à l'arrêt programmé.

(I770) = 00 0000 0000

(I773) = 00 XXXX XXXX

I772 20 I770 I790

I790 60 I773 I749

I749 46 I771 I799 (14)

I771 11 I770 I753

(14) I799 10 I770 I753

(I753) = Instruction de sortie

Notation

A : Argument donné en virgule flottante du type FLAIR et en double précision. Il est chargé en 8003 et 8002. Son exposant α doit être inférieur à 54.

$a = 2\pi$: Argument transformé en virgule fixe et réduit dans l'intervalle $(-\frac{\pi}{2}, \frac{\pi}{2})$

Limite de l'exposant

Elle est imposée par la réduction dans $(-2\pi, 2\pi)$.

$\frac{A}{2\pi} = N + F$ N:Partie entière qui est ignorée; F:Partie décimale qui seule est traitée.

Pour $\alpha \geq 50$, F est obtenu avec (70 - α) chiffres significatifs. Compte tenu du fait que le 18^e chiffre est entaché d'erreur, nous prenons α inférieur à 54, afin d'obtenir au moins 17 chiffres significatifs pour F

Etude de l'exposant. Transformation de l'écriture "virgule flottante" de l'argument en écriture "virgule fixe".

$\alpha \geq 54$: Arrêt programmé.

$\alpha < 42$: L'argument a au moins 9 zéros en tête. Si nous nous référons au développement en série : $\cos A = 1 - \frac{A^2}{2} + \dots$ Le terme $\frac{A^2}{2}$ a au moins 18 zéros en tête et par suite est négligeable devant l'unité (à fortiori pour les autres termes). La valeur de $\cos A$ est donc prise égale à 1. $\sin A = A - \frac{A^3}{6} + \dots$ Le terme $\frac{A^3}{6}$ a au moins 28 zéros en tête. La valeur de $\sin A$ est prise égale à A.

$50 \leq \alpha < 54$: La mantisse de A divisée par 2π et décalée de $(\alpha - 50)$ positions vers la gauche donne F en virgule fixe et cadré 0 en 8000. $42 \leq \alpha < 50$: La mantisse de A divisée par 2π et décalée de $|\alpha - 50|$ positions vers la droite donne F.

Modification de F pour ramener le calcul du cosinus à celui du sinus.

Si F est positif : $\cos 2\pi F = -\sin 2\pi(F - \frac{1}{4})$

Si F est négatif : $\cos 2\pi F = \sin 2\pi(F + \frac{1}{4})$

Nous posons : $F_0 = F$ pour le calcul d'un sinus.

$F_0 = F + \frac{\varepsilon_0}{4}$ pour le calcul d'un cosinus. ($\varepsilon_0 = \pm 1$ suivant le signe de F.)

Réduction de πF_0 en πx .

- F₀ positif $\varepsilon_1 = F_0 - \frac{1}{2}$ $|f_1| < \frac{1}{2}$ $\sin 2\pi F_0 = -\sin 2\pi f_1$

F₀ négatif $\varepsilon_1 = F_0 + \frac{1}{2}$

f₁ positif $f = f_1 - \frac{1}{4}$

f₁ négatif $f = f_1 + \frac{1}{4}$

avec $\varepsilon_1 = -1$ si $f_1 > 0$; $\varepsilon_1 = 1$ si $f_1 < 0$.

Finalement : $\sin A = \varepsilon_0 \sin 2\pi F_0 = \varepsilon_0 \varepsilon_1 \cos 2\pi f$ (ε_0 toujours égal à 1)

$\cos A = \varepsilon_0 \sin 2\pi F_0 = \varepsilon_0 \varepsilon_1 \cos 2\pi f$ ($\varepsilon_0 = -1$ si $F_0 > 0$; $\varepsilon_0 = 1$ si $F_0 < 0$)

Le changement de variable $X = 4f$ donne: $\cos 2\pi f = \cos \frac{\pi}{2} X = \sum_{n=0}^{\infty} K_n X^n$.

Détermination du signe du résultat

Le signe du résultat final peut être obtenu en faisant le produit de ε_0 et ε_1 . Il est plus simple d'opérer ainsi :

A chaque séquence pouvant entraîner un changement de signe du résultat, nous enregistrons un mot de contrôle.

en (0002)	Mot de contrôle :	01 0000 0000	$\varepsilon_0 = 1$	Calcul de $\sin A$
"	"	01 0000 0000	$\varepsilon_0 = 1$	Calcul de $\cos A$
"	"	-01 0000 0000	$\varepsilon_0 = -1$	Calcul de $\cos A$
en (0014)	Mot de contrôle :	01 0000 0000	$\varepsilon_1 = -1$	Calcul de $\sin A$
"	"	-01 0000 0000	$\varepsilon_1 = 1$	ou de $\cos A$

(Le fait d'enregistrer un mot de contrôle de signe contraire à celui de ε_1 , permet de remplacer le produit par une somme).

Si la somme des contenus des mémoires (0002) et (0014) est nulle, le résultat à même signe que $\cos 2\pi f$ (toujours positif). Si la somme est différente de zéro (quel que soit son signe), il faut changer le signe de $\cos 2\pi f$.

Le résultat final est obtenu en virgule flottante dans l'accumulateur et dans le pseudo-accumulateur (1700, 1715, 1730).

La première instruction du sous-programme se trouve en 0000 pour le calcul d'un sinus ou en 0001 pour le calcul d'un cosinus.

Mémoires de travail

à contenus_constants_

(0007) = 00 0000 0008
 (0028) = 17 9586 4800 (Partie droite de 2π)
 (0036) = 00 0000 0054
 (0037) = 62 8318 5307 (Partie gauche de 2π)
 (0047) = 00 0000 0004
 (0050) = 00 0000 0042
 (0060) = 40 0000 0000
 (0063) = 50 0000 0000
 (0065) = 25 0000 0000
 (0073) = -01 0000 0000
 (0116) = 00 0000 0050

Coefficients de l'approximation polynomiale de $\cos 2\pi f$.

(0149) = 01,0000 0000 (K_0)
 (0112) = -12 3370 0550 (K_2)
 (0145) = -13 6169 8261 (K_2)
 (0114) = 25 3669 5079 (K_4)
 (0152) = 01 0479 7153 (K_4)
 (0138) = -02 0863 4807 (K_6)
 (0080) = -83 3524 2170 (K_6)
 (0159) = 00 0819 2602 (K_8)
 (0094) = 74 8359 1999 (K_8)
 (0030) = -00 0025 2020 (K_{10})
 (0066) = -42 3599 5968 (K_{10})
 (0104) = 00 0000 4710 (K_{12})
 (0095) = 87 4480 7624 (K_{12})
 (0082) = -00 0000 0063 (K_{14})
 (0124) = -86 5610 9448 (K_{14})
 (0158) = 65 8238 4799 ($K_{16} \cdot 10^{-10}$)
 (0146) = -00 5125 1382 ($K_{16} \cdot 10^{-10}$)

à contenus_variables_

(0002) = Mot d'orientation si $\epsilon_4 < 42$ (dans l'étude de l'exposant)

Son contenu est :

15 0050 0155 (calcul de $\sin A$)
 60 0149 0003 (calcul de $\cos A$)

= Partie gauche de A dans le cas où $\sin A = A$.

= Instruction de décalages si $42 \leq \epsilon_4 < 54$

Son contenu est :

30 00| ϵ_4 - 50| 0014 si $42 \leq \epsilon_4 < 50$
 35 00(ϵ_4 - 50) 0014 si $50 \leq \epsilon_4 < 54$

Cette instruction est utilisée pour l'élimination de la partie entière " N " (s'il y a lieu) et pour le cadrage extrême gauche de la partie fractionnaire " F ".

= Mot de contrôle dans le changement de " F " en " F_0 ".

01 0000 0000 si $\epsilon_3 = 1$ dans le calcul de $\sin A$ ou $\cos A$.

-01 0000 0000 si $\epsilon_3 = -1$ dans le calcul de $\cos A$.

= Mot d'orientation pour la détermination du signe du résultat final.

Son contenu est :

20 0014 0130 si $\epsilon_3 \epsilon_4 = -1$
 69 0118 0134 si $\epsilon_3 \epsilon_4 = 1$

(0014) = Mot d'orientation si $42 \leq \epsilon_4 < 54$ (dans l'étude de l'exposant de l'argument)

Son contenu est :

69 0149 0076 (calcul de $\sin A$)
 46 0111 0012 (calcul de $\cos A$)

= Mot de contrôle dans la réduction de " F_0 " en " F ".

01 0000 0000 si $\epsilon_4 = -1$

-01 0000 0000 si $\epsilon_4 = 1$

= Partie droite de " F ".

= Partie droite de $\cos 2\pi f$ (s'il y a changement de signe)

Mémoires appartenant à l'ADP

(I705) = Instruction à exécuter après chargement du multiplicande en I7I5 et I730.

Son contenu est successivement :

60 0002 0107 après chargement de "f" en vue du changement de variable $X = 4f$.

69 0137 0122 après chargement de "X" pour le calcul de son carré.

65 0143 0101 après chargement de " X^2 " pour le calcul de $\cos \frac{\pi}{2} X$.

pseudo-accumulateur

(I700) = 00 0000 0000 (exposant du résultat)

$pp = \alpha\alpha$ si $\sin A = A$ ($\alpha\alpha < 42$)

$pp = 00$ si $\cos(2k+1)\frac{\pi}{2}$ ou $\sin k\pi$

$pp = 50 - (n-2)$ dans les autres cas, avec

"n": nombre de décalages à gauche pour éliminer les zéros en tête du résultat écrit en virgule fixe.

(I715) = $\alpha\alpha$ XXXX XXXX Partie gauche de l'argument lors de l'étude de son exposant.

= Partie gauche du multiplicateur (f, X, X^2) écrit en virgule fixe et cadrée extrême gauche chaque fois qu'il y a multiplication en double précision.

= 00 YYYY YYYY partie gauche de la mantisse du résultat final
Elle est égale à celle de l'argument si $\alpha\alpha < 42$
($\sin A = A$); elle est nulle si nous avons
 $\cos(2k+1)\frac{\pi}{2}$ ou $\sin k\pi$.

(I730) = XX XXXX XXXX partie droite de l'argument lors de l'étude de son exposant.

= Partie droite du multiplicande (f, X, X^2) chaque fois qu'il y a multiplication en double précision.

= YY YYYY YYYY partie droite de la mantisse du résultat final
Elle est égale à celle de l'argument si $\alpha\alpha < 42$;
nulle dans le cas $\cos(2k+1)\frac{\pi}{2}$ ou $\sin k\pi$.

(I717) = Partie gauche du diviseur ou du multiplicateur écrit en virgule fixe.

(I731) = Partie droite du diviseur ou du multiplicateur.

(I740) = Instruction de sortie du sous-programme.

(I821) = Instruction du sous-programme à exécuter après chaque opération faisant appel à l'ADP.

= 21 I715 I876 instruction propre à l'ADP qui est rechargée en fin de programme.

Instructions

0000 24 I740 0043 Instruction de départ pour le calcul de sinus. En fin de programme il y a renvoi à l'ADP, aussi nous chargeons la mémoire de sortie de l'ADP.

0043 69 0046 0049 (0046) = I5 0050 0155

Prévision du cas où $\sin A$ serait pris égal à A. ($\alpha\alpha < 42$)

0049 24 0002 0005 (0002) = Mot d'orientation si $\alpha\alpha < 42$

0005 69 0008 0011 (1) (0008) = 69 0149 0076

Chargement du mot de contrôle correspondant à $\epsilon_0 = 1$

0001 24 I740 0093 Instruction de départ pour le calcul de cosinus.

0093 69 0096 0099 (0096) = 60 0149 0003

Prévision du cas où $\cos A$ serait pris égal à 1 ($\alpha\alpha < 42$)

0099 24 0002 0055 (0002) = Mot d'orientation si $\alpha\alpha < 42$

0055 69 0058 0011 (1) (0058) = 46 0111 0012

Test du signe de F pour la détermination de F_0 .

(1) 0011 24 0014 0017 (0014) = Mot d'orientation si $42 \leq \alpha\alpha < 54$

posant.

0017	20 I730 0033	(I730) = XX XXXX XXXX
0033	67 8003 0041	
0041	24 I715 0018	(I715) = $\alpha\alpha$ XXXX XXXX
0018	35 0002 0025	Isolément de l'exposant $\alpha\alpha$.
0025	65 8003 0083	(8003) = 00 0000 00 $\alpha\alpha$
0083	I6 0036 0091	(0036) = 00 0000 0054
		Comparaison de $\alpha\alpha$ avec 54
0091	46 0044 0045	
0045	01 0024 9999	(2)
		ARRÊT programmé pour $\alpha\alpha \geq 54$.
(2) 0044	I5 0047 0051	(0047) = 00 0000 0004
		Comparaison de $\alpha\alpha$ avec 50.
0051	45 0004 0105 (3)	$50 \leq \alpha\alpha < 54$. Reprise plus loin.
0004	I5 0007 0061	(0007) = 00 0000 0008
		Comparaison de $\alpha\alpha$ avec 42.
0061	46 0002 0015 (4)	$42 \leq \alpha\alpha < 50$. Reprise plus loin.
		(0002) = Mot d'orientation ($\alpha\alpha < 42$)
		60 0149 0003 (Calcul de cos A)
0096	60 0149 0003 (5)	(0149) = 01 0000 0000
		Valeur de cos A en virgule fixe. Renvoi à la séquence d'entrée dans l'ADP pour la transformation en écriture "virgule flottante" et le chargement du pseudo-accumulateur.
		(0002) = Mot d'orientation ($\alpha\alpha < 42$)
		I5 0050 0155 (Calcul de sin A)
		La valeur de sin A = A. Chargement de A dans 8000 et dans le pseudo-accumulateur.
0046	I5 0050 0155	(0050) = 00 0000 0042 (Reconstitution de $\alpha\alpha$)
0155	20 I700 0053	(I700) = 00 0000 00 $\alpha\alpha$
0053	60 I715 0019	(I715) = $\alpha\alpha$ XXXX XXXX
0019	24 0002 0006	

0006 35 0002 0013 Chargement de zéros dans les positions occupées par l'exposant.

0013 30 0002 0069 (I715) = 00 XXXX XXXX
 0069 21 I715 0068 (0002) = $\alpha\alpha$ XXXX XXXX
 0068 60 0002 0057 (I730) = XX XXXX XXXX
 0057 I5 I730 I740 (I740) = Instruction de sortie.

Préparation de l'instruction de décalages pour obtenir "P" cadré extrême gauche en 8000.

(4) 0015 I6 8001 0023 (8002) = 00 0000 00($\alpha\alpha$ - 50) avec $\alpha\alpha$ inférieur à 50.
 0023 69 0026 0029 (6) (0026) = 30 0000 0014
 (3) 0105 69 0108 0029 (0108) = 35 0000 0014
 (8002) = 00 0000 00($\alpha\alpha$ - 50) avec $\alpha\alpha \geq 50$
 0029 35 0004 0039
 0039 22 0002 0056 (0002) = Instruction de décalages vers la droite ou vers la gauche de ($\alpha\alpha$ - 50) positions.

Division de la mantisse par 2 Π et détermination de "P".

0056 65 I730 0035 (I730) = Partie droite de l'argument.
 0035 I0 I715 0119 (I715) = " " gauche " "
 0119 35 0002 0075 Elimination de l'exposant.
 Séquence d'entrée dans l'ADP pour effectuer une division sur des nombres en virgule fixe et en double précision. Le dividende est en 8000
 Le diviseur 2 Π est chargé dans les mémoires I717 et I731. La première instruction dans l'ADP est prise en I874 afin de "shunter" les calculs préalables sur les exposants du dividende et du diviseur. L'instruction à exécuter après la division est chargée en I821. Elle détruit l'instruction propre à l'ADP contenue dans cette mémoire et ainsi il y a "shuntage" du calcul de l'exposant du quotient.

0075	69 0028 0031	(0028) = Partie droite de 2 Π .
0031	24 I731 0034	
0034	69 0037 0040	(0037) = Partie gauche de 2 Π .
0040	24 I717 0020	

0020 69 0002 0106

(0002) = Instruction de décalages qui sera exécutée dès l'obtention du quotient en virgule fixé.

0106 24 1821 1874

1874

A.D.P
division de mantisses

1821

30

(0002) xxx 0014

35

(1821) = (0002)

Elimination de la partie entière s'il

y a lieu et cadrage extrême gauche de F_0 .(8000) = F_0 cadré 0.

(0014) = Mot d'orientation

46 0111 0012 (Calcul de $\cos A$)Modification de F_0 pour remener le calculde $\cos A$ à celui de $\sin A$ et enregistré-

ment du mot de contrôle correspondant.

Test du signe de F_0 .

(0065) = 25 0000 0000

(8000) = $F_0 = F - \frac{1}{4}$ cadré 0.(0073) = -01 0000 0000 ($\varepsilon_0 = -1$)(8000) = $F_0 = F + \frac{1}{4}$ cadré 0.

(0014) = Mot d'orientation

69 0149 0076 (Calcul de $\sin A$)

Enregistrement du mot de contrôle.

(0149) = 01 0000 0000 ($\varepsilon_0 = 1$)(0002) = Mot de contrôle du signe de ε_0 .Réduction de F_0 en F_0^{10} et chargement du mot de contrôle du signe de ε_1

0156 46 0009 0010

(10)

0010 11 0063 0067 (11)

(0063) = 50 0000 0000

(10)0009 10 0063 0067

(8000) = F_1 cadré 0.

(11)0067 46 0120 0021

(12)

0021 11 0065 0071 (0065) = 25 0000 0000

0071 69 0149 0052 (13) (0149) = 01 0000 0000 ($\varepsilon_4 = -1$)

(12)0120 10 0065 0121

0121 69 0073 0052 (0073) = -01 0000 0000 ($\varepsilon_4 = 1$)

0052 24 0014 0117

(0014) = Mot de contrôle du signe de ε_4
(8000) = f cadré 0.

" F_0 " est chargé dans les mémoires 1715 et 1730 en vue de la multiplication pour le changement de variable.

0117 69 0022 0125

0125

Chargement du multi-

plicande en 1715-1730

(1705) = (0022)

Multiplicande = F

(Voir en fin de programme)

Détermination du signe du résultat final; suivant le signe, enregistrement de l'instruction à exécuter après l'obtention de $\cos 2nf$.

0022 60 0002 0107 (0002) = Mot de contrôle du signe de ε_0 0107 10 0014 0072 (0014) = Mot de contrôle du signe de ε_1

0072 44 0126 0027 (14)

La somme des mots de contrôle est différente de zéro : le résultat final est négatif.

0126 69 0079 0032 (15) (0079) = 20 0014 0130

Instruction conduisant au chargement de signe de $\cos 2nf$.

La somme des mots de contrôle est nulle: il y a aucune modification de $\cos 2nf$. Enregistrement de la première instruction de la séquence d'en-trée dans l'ADP pour la transformation d'écriture du résultat et pour son chargement dans le pseudo-accumulateur.

(14)0027 69 0003 0032 (0003) = 69 0116 0134

(15)0032 24 0002 0157 (0002) = Mot d'orientation.

Changement de variable $X = 4f$. Résolution des cas où l'argument est arc remarquable. Calcul de $\cos \frac{\pi}{2} X$ par $\sum_{n=0}^3 K_n X^n$ suivant le schéma de Horner.

0157 60 0060 0115 (0060) = 40 0000 0000

0115 69 0118 0122

0122	Séquence d'entrée dans l'A.D.P pour effectuer une multiplication.	1806
Multiplicateur = 40 0000 0000 (Voir en fin de programme)		
1806	A.D.P multiplication de mantisses	1821
0118 20 0014 0123		
0123 65 8003 0081		
0081 35 0001 0087		
0087 44 0141 0042 (16)		
La partie entière est non nulle : la valeur absolue de X égale 1. Nous avons les cas : $\cos(2k+1)\frac{\pi}{2}$ et $\sin k\pi$. Chargement de zéros dans l'accumulateur et dans le pseudo-accumulateur.		
0141 60 8002 0100		
0100 24 1700 0103		
0103 24 1715 0024		
0024 24 1730 1740		
La partie entière est nulle. Calcul de X^4		
(16)0042 30 0001 0150		
0150 60 8002 0059		
0059 15 0014 0074		
0074 35 0001 0131		
0131 69 0084 0125		
0125	Chargement du multiplicande en 1715-1730	1705
Multiplicande = X (Voir en fin de programme)		
0084 69 0137 0122		

0122	Séquence d'entrée dans l'A.D.P pour effectuer une multiplication.	1806
Multiplicateur = X (Voir en fin de programme)		
1806	A.D.P multiplication de mantisses	1821
0137 45 0090 0092 (17)		
0090 69 0143 0125		
0125	Chargement du multiplicande en 1715-1730	1705
Multiplicande = X^2 Les contenus des mémoires 1715-1730 ne sont pas détruits après une multiplication de mantisses. Nous n'aurons plus à charger X^2.		
(1705) = (0143)		
(0146) = Partie droite de K_{16}		
(la partie gauche est nulle)		
0101 69 0054 0122		
0122	Multiplication DP	1821
Nous regroupons sous cette présentation la séquence d'entrée en A.D.P et la multiplication des mantisses en double précision.		
(1821) = (0054)		
(0158) = Partie droite de K_{16}		
(la partie gauche est nulle)		
0054 15 0158 0113		
0113 69 0016 0122		

(8000) = X^2 cadré 0.
(1821) = (0137)
 $X^2 = 0$. Nous avons les cas : $\cos k\pi$ et $\sin(2k+1)\frac{\pi}{2}$. Reprise à l'addition de K_0 dans le calcul de $\cos\frac{\pi}{2}X$.

0122	0097	I5 0152 0110	Multiplificateur = $K_4 + K_{18} X^2$ cadré 0.
Multiplification DP	0110	I0 0114 0078	(0114) et (0152) = K_2
1821	0078	69 0132 0122	Multiplificateur = $K_4 + \dots K_{18} X^{14}$ cadré 0.
0016	0122		(1821) = (0132)
I5 0124 0129	Multiplification DP		Cadrage 1 pour additionner K_2
I0 0082 0038	1821		(0112) et (0145) = K_2
0038	0122		Multiplificateur = $K_2 + \dots K_{18} X^{16}$ cadré 1.
0122			(1821) = (0133)
Multiplification DP	0132	30 0001 0089	Cadrage 2 avec arrondi. (Ce cadrage est imposé par l'ADP pour la transformation en écriture "virgule flottante").
1821	0089	I5 0145 0153	(0149) = Partie gauche de K_0
0142	0151	I0 0104 0109	(la partie droite est nulle)
I0 0104 0109	0128	69 0133 0122	(8000) = cos $2\pi f$ cadré 2.
0109	0122		(0002) = Mot d'orientation
0062	0122		20 0014 0130 (changement du signe de cos $2\pi f$)
I5 0065 0077	Multiplification DP		(0014) = Partie droite de cos $2\pi f$
I0 0030 0085	1821		(8000) = $\epsilon_0 \epsilon_1 \cos 2\pi f$
0085	0122		(0002) = Mot d'orientation :
0122			69 0116 0134 (cos $2\pi f = \epsilon_0 \cos 2\pi f$)
Multiplification DP	0088	I5 0094 0102	Séquence d'entrée dans l'A.D.P pour la transformation en écriture
1821	0102	I0 0159 0064	"virgule flottante" du résultat final et pour son chargement dans le pseudo-accumulateur.
0064	0122		(0003 69 0116 0134)
0122			(0116) = 00 0000 0050
Multiplification DP	0127	I5 0080 0135	L'exposant du résultat sera : $\beta\beta = 50 - (n-2)$
1821	0135	I0 0138 0144	(n = nombre de décalages à gauche pour éliminer les zéros en tête).
0144	0122		
0122			
Multiplification DP	0138		
1821	0144		
0122			
Multiplification DP			
1821			

0097	I5 0152 0110	Multiplificateur = $K_4 + K_{18} X^2$ cadré 0.
0110	I0 0114 0078	(0114) et (0152) = K_2
0078	69 0132 0122	Multiplificateur = $K_4 + \dots K_{18} X^{14}$ cadré 0.
0122		(1821) = (0132)
Multiplification DP		Cadrage 1 pour additionner K_2
1821		(0112) et (0145) = K_2
0132	30 0001 0089	Multiplificateur = $K_2 + \dots K_{18} X^{16}$ cadré 1.
0089	I5 0145 0153	(1821) = (0133)
0153	I0 0112 0128	Cadrage 2 avec arrondi. (Ce cadrage est imposé par l'ADP pour la transformation en écriture "virgule flottante").
0128	69 0133 0122	(0149) = Partie gauche de K_0
0122		(la partie droite est nulle)
Multiplification DP		(8000) = cos $2\pi f$ cadré 2.
1821		(0002) = Mot d'orientation
0133	31 0001 0092	20 0014 0130 (changement du signe de cos $2\pi f$)
(17) 0092	I0 0149 0002	(0014) = Partie droite de cos $2\pi f$
		(8000) = $\epsilon_0 \epsilon_1 \cos 2\pi f$
		(0002) = Mot d'orientation :
		69 0116 0134 (cos $2\pi f = \epsilon_0 \cos 2\pi f$)
		Séquence d'entrée dans l'A.D.P pour la transformation en écriture
		"virgule flottante" du résultat final et pour son chargement dans le pseudo-accumulateur.
		(0003 69 0116 0134)
		(0116) = 00 0000 0050
		L'exposant du résultat sera : $\beta\beta = 50 - (n-2)$
		(n = nombre de décalages à gauche pour éliminer les zéros en tête).

0134 24 I700 0154

0134 59 0036 0140

0140 24 I82I 800I

I82I

A.D.P

I740

(0036) = 2I I7I5 I876

Instruction initiale de l'ADP contenue dans la mémoire I82I. Cette instruction avait été détruite afin de "shunter" le calcul sur les exposants.

Le résultat en virgule fixe est transformé en virgule flottante du type FLAIR. Il est chargé dans le pseudo-accumulateur et dans 8000.

(I740) = Instruction de sortie.

Chargement du multiplicande contenu dans 8000 dans les mémoires I7I5 I730. 800I contient l'instruction à exécuter après chargement.

0125 24 I705 0048 (I705) = Instruction à exécuter après chargement.

0048 2I I7I5 0098

0098 20 I730 I705

Séquence d'entrée dans l'A.D.P pour effectuer une multiplication sur des nombres en virgule fixe et en double précision.

Le multiplicande est en I7I5-I730. Le multiplicateur initialement en 8000 est chargé dans les mémoires I7I7-I73I. La première instruction dans l'ADP est prise en I806 afin de "shunter" les calculs préalables sur les exposants du multiplicande et du multiplicateur. L'instruction à exécuter après la multiplication est chargée en I82I : il y a destruction de l'instruction propre à l'ADP contenue dans cette mémoire et par suite "shuntage" du calcul de l'exposant du produit.

0122 24 I82I 0136

0136 2I I7I7 0147

0147 20 I73I I806

Entrée dans l'A.D.P

ANNEXE II

BIBLIOGRAPHIE

SOMMAIRE

- (1) DENMAN Computer Generation of Optimized Subroutines- Journal of the Association for Computing Machinery, Janvier 1961. (P. 104 à 116)
 - (2) HASTINGS Approximations for Digital Computers- Princeton University Press, 1955.
 - (3) KOGBETLIANTZ Generation of elementary functions- Mathematical Methods for Digital Computers, Wiley, 1960. (Première partie)
Conférences- Septembre 1956.
 - (4) LANCE Numerical methods for high speed Computers- Iliffe 1960. (Chap. 2)
 - (5) LANCZOS Applied Analysis-Prentice Hall, 1956.
 - (6) VIONNET Approximation de Tchebycheff des fonctions- Chiffres-2, 1959 et 1960.
 - (7) BELHACHE Thèse- Arithmétiques pour nombres réels, double et triple précisions et pour nombres complexes, simple et double précisions, 1964.
 - (8) Tables of Chebyshev Polynomials- National Bureau of Standards' Applied Mathematics Series, 1952.
-

Deuxième Thèse

Propositions données par la Faculté :

EMPLOI DE SERIES ASYMPTOTIQUES
POUR LE CALCUL NUMERIQUE DES FONCTIONS

VU

Le Directeur de Thèse,

G. BRILLOUET

VU

Le Président du Jury,

G. BRILLOUET

VU et APPROUVE

NANTES, le 2 JUIN 1965

Le Doyen de la Faculté des Sciences,

G. BRILLOUET

VU et PERMIS d'IMPRIMER

NANTES, le 3 JUIN 1965

*Le Recteur de l'Académie de Nantes
Président du Conseil de l'Université,*

M. SCHMITT

